Australian Government
Department of Defence
Defence Science and
Technology Organisation

Video Moving Target Indication in the Analysts' Detection Support System

Ronald Jones, David M. Booth and Nicholas J. Redding

Intelligence, Surveillance and Reconnaissance Division
Defence Science and Technology Organisation

DSTO–RR–0306

ABSTRACT

This report presents a review of a video moving target indication (VMTI) capability implemented in the Analysts' Detection Support System (ADSS). The VMTI subsystem has been devised for video from moving sensors, in particular, but not exclusively, airborne urban surveillance video. The paradigm of the moving sensor, which is a typical scenario in defence applications (*e.g.*, UAV surveillance video), poses some unique problems as compared to the stationary sensor. Our solution to these problems draws on a number of algorithms from the computer vision community, and combines them in a novel system. It will provide positional and size information for any moving targets in a given video sequence, on a frame by frame basis. Moreover, given suitable parallel non-specialised hardware, the system allows a near real time solution to VMTI in ADSS.

Published by

Defence Science and Technology Organisation
PO Box 1500
Edinburgh, South Australia 5111, Australia

Telephone: (08) 8259 5555
Facsimile: (08) 8259 6567

AR No. 013-600
April, 2006

APPROVED FOR PUBLIC RELEASE

Video Moving Target Indication in the Analysts' Detection Support System

EXECUTIVE SUMMARY

This report presents a review of the video moving target indication (VMTI) subsystem currently implemented within the Analysts' Detection Support System (ADSS). The ADSS was originally developed to assist in the exploitation of synthetic aperture radar (SAR) imagery, but developments over the past several years have facilitated effective processing of streaming video data. New algorithms have been incorporated to support urban surveillance from airborne and ground-based platforms, in particular VMTI. VMTI for a static camera has been well researched and reported in the literature over the past decade, and there are a number of excellent COTS products available to detect and analyse motion in video from static cameras (*e.g.*, the products offered by the Australian companies *Clarity Visual Intelligence* and *Sentient Software*). VMTI for moving cameras however is considerably less mature. We saw this technology gap as an opportunity to develop our own video processing algorithms within ADSS, for important applications such as VMTI for Unmanned Aerial Vehicle (UAV) surveillance.

The paradigm of the moving sensor poses some unique problems as compared to the stationary sensor because, relative to the camera, everything in the scene appears to be moving. The motion of the actual targets must then be distinguished from the global motion in the scene. Our solution draws on a number of algorithms from the computer vision community, and combines them in a novel system. In particular, we leverage existing algorithm development we have in shape from motion, *e.g.*, feature tracking and outlier removal, and combine it with established work for the static camera scenario, *e.g.*, background modelling and frame differencing. The solution provides positional and size information for any moving targets in a given video sequence, on a frame by frame basis. Moreover, given suitable parallel non-specialised hardware, the system allows a near real-time solution to VMTI. A VMTI system needs to run at a near real-time rate to be of any operational value in the field; we are not aware of any existing real-time VMTI system matching our performance capabilities. We compare two other VMTI systems with ours and provide a comparative analysis.

The technique reported here won the "Best Paper" award when it was presented in a shortened form at the recent Digital Image Computing: Techniques & Applications Conference in Cairns, December 2005.

Authors

Ronald Jones
Intelligence, Surveillance and Reconnaissance Division

Ronald Jones received a B.Sc. and Ph.D. in Physics from Victoria University of Wellington and Monash University, Melbourne, in 1990 and 1994, respectively. Since completing his PhD, he has worked as a Research Scientist at the CSIRO, Division of Mathematical and Information Sciences, and Proteome Systems Ltd, a BioTech company based in Sydney. He joined DSTO as a Senior Research Scientist in 2004. His research interests currently include linear and nonlinear image processing techniques. In 2006 he began a new career by entering a postgraduate medical degree at Flinders University to become a medical practitioner.

David M. Booth
Intelligence, Surveillance and Reconnaissance

David Booth has a BSc(Hons)degree in Computer Science (1984), an MPhil in Remote Sensing (1986), and a PhD in Pattern Recognition (1991). He has been engaged in scientific research for The Ministry of Defence (UK) since 1986, mainly in the imagery exploitation field. During that time he led the ISTAR Imagery Exploitation Research Team at DERA Malvern for five years, and subsequent to that, the Image Understanding Project Team at DSTL Malvern. He is currently on long term detached duty at DSTO until June 2006. He is a Chartered Engineer, a Chartered Information Technology Professional, and a Fellow of the British Computing Society.

Nicholas J. Redding
Intelligence, Surveillance and Reconnaissance Division

Nicholas Redding received a B.E. and Ph.D. in electrical engineering all from the University of Queensland, Brisbane, in 1986 and 1991, respectively. In 1988 he received a Research Scientist Fellowship from the Australian Defence Science and Technology Organisation (DSTO) and then joined DSTO in Adelaide as a Research Scientist after completing his Ph.D. in artificial neural networks in 1991. In 1996 he was appointed as a Senior Research Scientist in the Microwave Radar Division (now Intelligence, Surveillance and Reconnaissance Division (ISRD)) of DSTO. Since joining DSTO he has applied image processing techniques to the automatic classification of ionospheric data, and more recently has researched target detection (both human and algorithmic) in synthetic aperture radar (SAR) imagery. In 2001 he returned from a one and a half year posting to the UK's Defence Evaluation and Research Agency where he continued the development of a suite of target detection algorithms for SAR imagery and researched new algorithms in SAR image forming using the circular Radon transform. In 2004 he was appointed Head of the Image Analysis & Exploitation Group within the ISRD.

Contents

1 Introduction

Video surveillance is an essential and commonly used mechanism for protecting vital infrastructure and improving situational awareness. However, manual exploitation of surveillance video can be such an intensive activity that often its only practical role is either as a visible deterrent or for post-mortem analysis following a particular event. Such utility is unacceptable when the event requires interdiction before there is loss of life or infrastructure. The instigation of a real-time response can be facilitated by using an active video surveillance approach, where automatic processing of multiple video streams draws the attention of analysts to suspicious activity, leaving the vast majority of benign imagery to pass unchecked by the human analyst.

This report presents a review of the video moving target indication (VMTI) subsystem currently implemented within the Analyst's Detection Support System (ADSS). The ADSS is a flexible processing engine developed to assist the imagery analyst to detect targets in all-source surveillance imagery. It provides the means of structuring a hierarchy of algorithms which, when applied to the data, makes progressively refined decisions on the locations of targets. The ADSS was originally developed to assist in the exploitation of synthetic aperture radar (SAR) imagery [46], but recent infrastructure developments have facilitated effective processing of streaming video data and new algorithms have been incorporated to support urban surveillance from airborne and ground-based platforms.

VMTI for a static camera (also known as motion segmentation) has been well researched and reported in the literature, and there are a number of excellent commercial products available to detect and analyse motion in video from static cameras (*e.g.*, the products offered by the Australian companies *Clarity Visual Intelligence* and *Sentient Software*). The standard approach adopted is to compare the current frame with a suitable background model constructed from the previous set of frames in the sequence [26]. A significant difference indicates a change in the scene has occurred, from which it can be inferred that there is motion in the scene. The motion can then be tracked through the subsequent frames using a tracking algorithm of choice, *e.g.*, particle filters [48] or simple heuristical methods based on shape characteristics and temporal integration (reported herein). The background model is continually updated with the current frame, excluding those pixels in the frame deemed to be part of moving targets.

In the case of a moving camera however, the situation is significantly more complex because, relative to the camera, everything in the scene appears to be moving. The motion of the actual targets must then be distinguished from the global motion in the scene. The problem can be addressed using one of a number of approaches, *e.g.*, background model based, correspondence based and optic flow based [31]; see also [11], [25] and [29] for other approaches to the problem. We have explored and implemented all of these approaches in ADSS, however the approach we report herein is based on background modelling. We have found that this approach is robust to sensor noise and typically yields complete, high quality object segmentations. Moreover, it provides persistent target detection should the object stop moving momentarily. This in turn provides a high quality input to the tracking phase.

This report will proceed as follows. In the following section, we provide a discussion and background theory of VMTI for the moving sensor scenario. In a subsequent report, we will describe the VMTI subsystem as it is implemented in ADSS, and provide details on the individual modules that make up the subsystem. Section 3 provides some results from our VMTI work, illustrating some of the issues that arise for the moving sensor scenario. In Section 4, we present the

results of a comparative analysis between the VMTI subsystem we have developed and two other VMTI systems available: ARIA, a VMTI approach developed by Caprari [12] and implemented in the MATLAB programming language, and a COTS product developed by an Australian software company. Finally, we conclude with some remarks in Section 5.

2 Moving Target Indication

2.1 Related Work

The automated detection and tracking of moving targets using video technology situated on an airborne platform has received comparatively little attention, mainly due to the lack of available imagery and the sensitivity of defence research.

Much of the earlier work focused on FLIR (forward looking infrared) sensors but many of the approaches were later applied in the visible band. The FLIR fraternity believe they have the more difficult problem: low signal-to-noise, non-repeatability of target signature, competing background clutter, lack of *a priori* information, high ego motion, and weather induced artifacts [62] .

Strehl and Aggarwal's [56] approach is based on the subtraction of registered frames followed by blob extraction and association etc. The frame-to-frame mapping resulting from ego-motion is modelled as affine, and determined by registering the frames using robust, multiscale matching of the entire frames i.e. it is not feature based. The affine model is unable to capture the skew, pan and tilt of the planar scene.

Shekarforoush and Challappa [53] combine sensor stabilisation and detection into a single stage, however, by essentially using the targets as feature points for registration, the process is dependent on persistent, high contrast targets and a relatively benignly textured background.

Some other authors made assumptions about the target characteristics or platform motion that weakened their proposals. For example, Braga-Neto and Goutsias [8] (who used morphological operators) assume that target sizes remain constant over time, that they exhibit high contrast with their surroundings, and that ego-motion is small. Davies *et al.* [20] proposed a Kalman filter-based target tracker but made strong assumptions about target motion and assumes no ego-motion.

In the visible band, the approach adopted by Yilmaz *et al.* [62] is relatively unconstrained: it accommodates high global motion; changes in target signature, and the targets need not move with constant velocity or acceleration. Significant (global) ego-motion is handled using the multi-resolution framework proposed by Irani and Anandan [32] if warranted. Targets are detected using an image filtering and segmentation scheme. The target distributions are then modelled, and the motion between this and the subsequent frame is determined by finding the translation vector in image space that minimises the probabilistic distance between model and candidate. This, mean-shift, approach was originally proposed by Comaniciu *et al.* [17] and has been used widely, particularly in ground-based surveillance applications when targets are large and have inconstant signatures. Mean shift tracking was used by Ali and Shah [1] for tracking vehicles in airborne (visible) video. The authors claim good performance when tracking targets larger than 100 pixels in area, and we suggest that it is the mean-shift tracker part of their system that is responsible

for this soft constraint. Ali and Shah remove ego motion using a feature-gradient descent hybrid registration scheme, the aim being to exploit the robustness of feature-based methods and the accuracy of gradient descent approaches. Moving targets are detected by accumulation of differences between a frame and its n neighbours, followed by histogramming of the logs. Large peaks correspond to background while smaller peaks are targets. Here too, the targets must be large to be sure of detecting a corresponding peak in the histogram. Target segmentation is achieved using level sets [63, 65].

Cheng and Butler [13] (Sarnoff Corporation) describe a video segmentation algorithm based on combining three outputs: a moving object detector / segmenter (based on background modelling), an unsupervised segmenter (based on local image properties), and a supervised segmenter (trained to distinguish between object classes, such as vehicle, tree, house). The three are combined based on their semantic meaning e.g. a vehicle can move, a house cannot.

Cohen and Medioni's initial strategy [14] was to register consecutive frames by minimising the least squares criterion subject to an affine transformation model (to remove ego-motion), followed by the detection of moving objects by detecting anomalies in the normal component of the residual flow. Later, stabilisation became based on feature tracking. After detection, the objects were tracked using a dynamic template, and trajectories extracted using a graph searching algorithm [15]. In a subsequent publication, Cohen and Medioni emphasised the unification of the stabilisation and detection stages [16], and Bremond and Medioni [7] describe an adjunct to the Cohen and Medioni system for recognising behavioural scenarios (based on the use of Petri Nets).

Dale *et al.* [18] describe a number of video exploitation algorithms that have been implemented on the ADEPT hardware. They parameterise the global motion field by a planar perspective model, which is capable of capturing translation, rotation, scale, shear and perspective projection. It is derived by tracking salient (i.e. strong, persistent and consistent) features. Thus, the tracking achieves a level of robustness though not necessarily by adopting a statistically rigorous approach such as RANSAC. Scene content determines the complexity of the global warp (adaptively), that is, from translation through to perspective projection. The targets are detected by image subtraction and, as such, the resulting segmentations are of poor quality and of a nature that is difficult to predict.

Removal of ego motion by global registration is an often used first stage in detecting moving targets. The aim is normally to register corresponding ground features. However, in addition to global motion, camera motion produces parallax artifacts, that is, the appearance of independent motion in objects that are fixed but elevated in comparison with the ground, and these can be indistinguishable from moving targets. In an attempt to identify some of these artifacts, Yalcin *et al.* [60] describe a flow-based approach which partitions a frame into foreground and background occlusion layers using an EM-based motion segmentation. Dong and Jinwen [22] prefer to remove parallax artifacts and propose a morphological procedure to do so. Reliable georeferencing is also becoming increasingly practical through calibrated camera kinematics and precision registration of video frames to reference imagery [59]. Planar-plus-parallax use a more sophisticated model of image motion which can capture the dominant planar motion as well as lines along which the residual parallex motion is expected [50]. Burns [9] examines techniques based on georeferenced object motion relative to the trajectory of the camera, as well as a new method of classifying objects and events using features extracted from georeferenced trajectories.

The reader can see that this airborne surveillance problem has essentially three stages: stabilisation (or registration [35]) , detect and track. Each stage has many potential types of solution,

the selection of which will be influenced by overall tracking performance, execution time, and robustness to unfavourable scene content or sensing conditions. In general, it is difficult to draw hard and fast rules regarding the suitability of one class of component algorithm over that of another, but in building a system we aim to develop components which complement one another, in particular, early modules should produce outputs with characteristics and a quality that suits later modules, and that later modules are robust in any shortcoming in the former. We also look to rely on a small number of parameters, and that those that are necassary should have a sound statistical basis and have meaning to the human operator.

This report describes the development of our system, different elements and slants on which have been published widely [47, 36, 6]. The tracking element will be the subject of a future report but the reader is referred to Jones *et al.* [36] for the application of a particle filter, and the more recent application of a Probabilistic Multi-Hypothesis Tracker (PMHT) is described by Davey [19].

2.2 Overview

The VMTI system is based on the assumption that pixels which compose a moving object will usually manifest themselves as statistical outliers from a model of the scene which has been constructed over an extended period of time. The basic strategy when dealing with a moving camera is to apply a video registration process to each frame in the sequence to remove the effects of the camera motion, thus allowing background modelling and outlier identification techniques to be applied.

When constructing a background model from a video sequence, an important notion is that of the temporal window used, or the set of frames from which the background model is constructed. For a static camera, this is usually the entire set of image frames that have been acquired up to the current point in time (though the model is likely to adapt over time). When the camera is moving however, the length of the temporal window is determined by the following considerations:

- It should be sufficiently short that all frames within the temporal window overlap spatially by a significant amount.
- It should be sufficiently short that image differences introduced by changes in viewing geometry as the camera moves through the scene are acceptable for constructing a background model. These differences generally increase with distance between camera positions, and cannot be entirely removed by a registration process based on a simple parametric registration model such as a global affine or projective transform.
- It should be of long enough duration to avoid a contribution to the background model being made by moving targets, at least to the extent that they do not impact on detection performance.

These competing considerations relate directly to the speed of the sensor, its proximity to the scene and the size and speed of the targets in the scene. A suitable choice for temporal window length is therefore dependent on the type of imagery at hand. We have found for our applications that a length of 100 frames often yields an acceptable background model while minimising the effects of perspective errors (based on standard definition video with 24 frames per second).

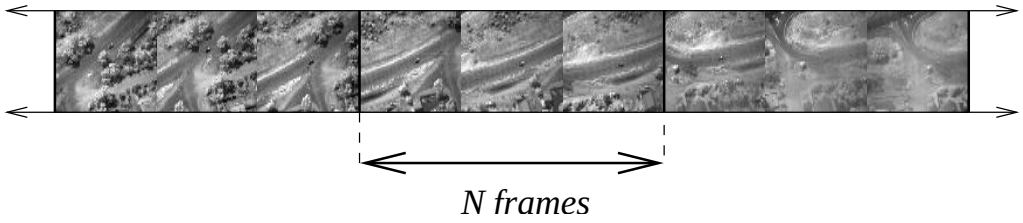

Figure 1: *The video sequence is divided into blocks each with N frames. VMTI is carried out within each block separately, independently of other blocks.*

Given the notion of a limited temporal window, our approach is then to divide the video sequence into blocks of N frames, where N is the length of the temporal window, as shown in Fig. 1. VMTI is carried out in each block separately: the N frames in the block are registered to the central frame in the block; a background model is formed from the stack of N registered frames; and frame differencing within the block carried out for each frame to produce the VMTI results. The key observation in the approach is that a valid background model can be formed from any contiguous set of frames around f, and not necessarily the frames directly previous to f in the sequence. Each frame f in the block may therefore be compared to a background model constructed from the N frames within that block. This approach allows for the efficient and computationally simple algorithm of division into blocks as shown in Fig. 1.

It is important to note that the approach also allows a parallel implementation of VMTI, as the independent processing of blocks can be carried out on separate processors. The VMTI results may then be recombined seamlessly into a continuous sequence. We may exploit this fact to devise a near real-time solution to VMTI on relatively cheap non-specialised hardware. The parallel implementation does however entail an inherent lag: Assuming sufficient processors for real time processing, the lag is $P \times N$ frames, where P is the number of processors. Further details on performance of the algorithm are given in a comparative analysis in Section 4.

2.3 The Registration Process

The ADSS includes a number of well known image registration techniques that could be applied to video data, including wavelets [42] and optical flow [31]. In this application, the favoured registration algorithms are the hierarchical, region based (or more precisely image based) correlation technique [45], and more so, the feature-based technique described below.

We report here our method of video registration based on a feature tracking algorithm of Kanade, Lucas and Tomasi [54], known as “KLT”, which has shown consistently good results over a wide range of video imagery. It is a mature feature tracking method that is well established in the computer vision community for tracking features in video sequences for the purpose of determining structure from motion [28]. In the KLT algorithm, small features such as corner points are extracted and tracked based on a “corneredness” measure, derived from the eigenvalues of the autocorrelation of the image intensities within a window, and the use of a dissimilarity measure to determine the affine transformation. For our purposes, the tracked features are simply used as control points to which a frame-to-frame parametric registration model is fitted (either affine or projective) and used to warp each frame to the common frame of reference. From here background modelling and frame differencing can be applied to yield the larger objects that we

wish to segment and track, *e.g.*, people, cars and other objects. This was a good opportunity to leverage an existing technology implemented in ADSS for shape from motion, by utilising the modular design and framework of the ADSS architecture.

The registration process may be formalised as follows: given a set of feature points $\mathbf{P}$ in frame $\mathbf{F}$ that have been tracked and correspond to a second set of points $\mathbf{P'}$ in some other frame $\mathbf{F'}$, they may be related using via the matrix equation $\mathbf{P'} = \mathbf{PA}$. Here $\mathbf{A}$ is 3×3 matrix capturing the parameters of either an affine or projective transform, and $\mathbf{P}$ and $\mathbf{P'}$ are matrices of points stored in rows as parametric triplets, $(x, y, 1)$. The equation can be solved for $\mathbf{A}$ using a simple least squares fit to obtain the best fit solution for either the affine or projective case. We may also obtain a measure of the reprojection error, or the degree to which $\mathbf{A}$ fits the data $\mathbf{P}$ and $\mathbf{P'}$, by computing the average difference:

$$\mathbf{A}_e = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{P}'_i - (\mathbf{PA})_i\|, \tag{1}$$

where i is an index into the N points of the point sets $\mathbf{P'}$ and $\mathbf{PA}$.

The matrix $\mathbf{A}$ is used as a model of the mapping function relating the images $\mathbf{F}$ and $\mathbf{F'}$, denoted herein by the relation $\mathbf{F'} = \mathbf{FA}$. In practice however, the registration of image $\mathbf{F'}$ to the domain of $\mathbf{F}$ can then be done in either the forward or backward direction. In the forward direction, each pixel in $\mathbf{F'}$ is directly transformed to the domain of $\mathbf{F}$ using the estimated mapping function, $\mathbf{A}$. However, due to rounding and discretisation errors, this can lead to holes and/or overlapping pixel values. For this reason, the backward direction is usually preferred and is the method we adopt in our work. In this case, the inverse transform $\mathbf{A}^{-1}$ is computed and each coordinate in $\mathbf{F}$ is mapped to the domain of $\mathbf{F'}$, from which a pixel value is computed from $\mathbf{F'}$ by interpolation. There are various options for interpolation method, such as nearest neighbour, bilinear, quadratic and least squares; we prefer bilinear interpolation because it is simple, efficient and yields acceptable results.

Figure 2 shows an example taken from a scene of urban surveillance. The two frames at the top of the figure are 50 frames apart, with feature points superimposed in white. The frame at the bottom of the figure is the registered version of the top frame in the sequence, using an affine registration model. The KLT algorithm has been applied to the sequence tracking $N = 200$ features. Features that are lost are immediately replaced by new features so that the maximum number of features is represented in any given frame. It is not always possible to find the maximum N features however, in particular in cases for large N and/or frames with relatively few features. The number of feature correspondences between any two given frames generally falls well short of N, as points are continually lost as the distance between frame $\mathbf{F}$ and frame $\mathbf{F'}$ increases.

2.3.1 The Kanade, Lucas and Tomasi (KLT) Algorithm

For the KLT algorithm, we use an implementation written by Birchfield [4] that we have modified slightly to improve its' speed. The algorithm as implemented works as follows.

The central idea behind the KLT algorithm is to track features across successive frames of the video sequence. The features themselves are defined in a manner that increases their likelihood of being tracked across the frames — hence the theme of "good features to track" in the titles of published papers in the area. Candidate features are computed from the smallest of the eigenvalues

Figure 2: *Feature tracking for video registration. Top and middle: two frames from the sequence. Bottom: Top frame registered to the middle frame using affine model fitting to the tracked features.*

λ_1, λ_2 of the matrix of gradients at each pixel $\mathbf{x}$ of a frame,

$$\mathbf{Z}(\mathbf{x}) = \begin{pmatrix} g_x^2(\mathbf{x}) & g_x(\mathbf{x})\, g_y(\mathbf{x}) \\ g_x(\mathbf{x})\, g_y(\mathbf{x}) & g_y^2(\mathbf{x}) \end{pmatrix} \tag{2}$$

where $g_x(\mathbf{x})$, $g_y(\mathbf{x})$ is the image derivative at a pixel $\mathbf{x}$ in the x and y direction, respectively. The matrix $\mathbf{Z}$ is intimately related to the equation solved during the tracking across frames as we will see in a moment. The pixels in the first frame are ranked in descending order of the smaller eigenvalue for each pixel, $\min(\lambda_1(\mathbf{x}), \lambda_2(\mathbf{x}))$, and the top N pixels in the list are selected as features for tracking. The list is winnowed before the selection occurs so that the feature points are not too closely spaced. A simple but significant speedup is obtained if particularly small eigenvalues are dropped from the sort.

Once the feature points have been selected, the next step is to determine their location in subsequent frames. This is done by firstly modelling the transformation between each frame as a displacement, where the dissimilarity between two windows is expressed as [54]

$$\epsilon = \sum_{\mathbf{x} \in W} \left(F_{n+1}(\mathbf{x} + \mathbf{d}) - F_n(\mathbf{x}) \right)^2 \, d\mathbf{x} \tag{3}$$

where W represents a window around the pixel $\mathbf{x}$ with value $F_n(\mathbf{x})$ in the n-th image $\mathbf{F}_n$ which moves to the position $\mathbf{x} + \mathbf{d}$ in the $n+1$-th image $\mathbf{F}_{n+1}$ in the sequence. Then the location of a particular feature in the next frame can be found by minimising the dissimilarity measure (3).

By taking a Taylor series expansion of (3) about the point $F_{n+1}(\mathbf{x})$ truncated to the linear term and using vectorisation operators and Kronecker products for the resulting equations [54], (3) can be approximated as

$$\mathbf{T}\, \mathbf{d} = \mathbf{e}, \tag{4}$$

where

$$\mathbf{T} = \sum_{\mathbf{x} \in W} \mathbf{Z}(\mathbf{x}),$$

$$\mathbf{d} = \begin{pmatrix} d_x \\ d_y \end{pmatrix},$$

$$\mathbf{e} = \sum_{\mathbf{x} \in W} \left(F_n(\mathbf{x}) - F_{n+1}(\mathbf{x} + \mathbf{d}) \right) \begin{pmatrix} g_x(\mathbf{x}) \\ g_y(\mathbf{x}) \end{pmatrix}$$

and d_x, d_y are the displacements in the x and y directions, respectively. To find the displacement for each feature, we repeatedly solve (4) for $\mathbf{d}$, with the starting assumption that the displacement of the feature between images is zero, until the change in displacement from one iteration to the next is small. If the iteration limit is exceeded, or the determinant of $\mathbf{T}$ is too small, then the feature is deemed to be lost and is dropped.

After the translation for each feature has been found, a consistency check on the feature is undertaken by transforming back to the very first frame in which it was detected, and if it has now become too dissimilar (via (3)) it is deemed to lost. We found that a simple translation model was

sufficient for our purposes, although an affine consistency check is also catered for in the code by Birchfield [4]. Note that bilinear interpolation is used to resample the images to accommodate sub-pixel translations. The implementation employs a multi-resolution approach to the tracking to provide good initial conditions for the displacement at the higher resolutions.

When the number of features has fallen below a threshold, a process is initiated that adds new features to the existing ones that have been successfully tracked.

2.3.2 The Iterative Registration Strategy

An important issue to be considered in the registration process is the compound error that occurs when sequentially registering frames in a video sequence. For example, the following 'cascaded' registration strategy could be used to register a sequence of N frames $\mathbf{F}_i$:

$$\mathbf{F}_2 = \mathbf{F}_1\mathbf{A}_{1,2}$$

$$\mathbf{F}_3 = \mathbf{F}_2\mathbf{A}_{2,3} = \mathbf{F}_1\mathbf{A}_{1,2}\mathbf{A}_{2,3}, \ldots,$$

$$\mathbf{F}_N = \mathbf{F}_{N-1}\mathbf{A}_{N-1,N} = \mathbf{F}_1\mathbf{A}_{1,2}\mathbf{A}_{2,3}\ldots\mathbf{A}_{N-1,N} \tag{5}$$

Here $\mathbf{A}_{i,i+1}$ is the transformation (*e.g.*, affine or projective) required to register frame $\mathbf{F}_i$ to frame $\mathbf{F}_{i+1}$. For a sequence of N frames then, registering frame $\mathbf{F}_1$ to frame $\mathbf{F}_N$ requires a cascade of $N-1$ separate transforms $\mathbf{A}_{i,i+1}$, $i = 1\ldots N-1$. Each of these transforms will involve some error in calculation, and cascaded them will produce a compounded error that can rapidly become unacceptable for the purposes of VMTI.

For example, Fig. 3a shows a plot of the reprojection error $\mathbf{A}_e$ (defined in Eq. 1) versus frame number for a simulated data set. Here the data set consisted of 200 random (x, y) coordinates with values $x, y \in [1, 50]$. An arbitrary affine transform was specified and applied iteratively to the data set 100 times to simulate a fixed camera motion for 100 frames. Gaussian noise of mean zero and variance one was then added to the data set to simulate the error in feature position measurement. As can be seen from the figure, the registration error increases almost linearly with frame, until eventually it is almost ten pixels. The registration process by this stage will produce frame registrations with an unacceptably high degree of error and will be unsuitable for the purposes of background modelling.

Using the method of feature tracking however, there is a simple way to significantly mitigate the effects of compound error, because we do not need to rely on a cascade of affine transforms. If we consider again registering a sequence of N frames $\mathbf{F}_i$, if we have successfully tracked all points through the N frames then we can apply the following 'non-cascaded' registration strategy:

$$\mathbf{F}_2 = \mathbf{F}_1\mathbf{A}_{1,2}$$

$$\mathbf{F}_3 = \mathbf{F}_2\mathbf{A}_{2,3} = \mathbf{F}_1\mathbf{A}_{1,3}, \ldots,$$

$$\mathbf{F}_N = \mathbf{F}_{N-1}\mathbf{A}_{N-1,N} = \mathbf{F}_1\mathbf{A}_{1,N}$$

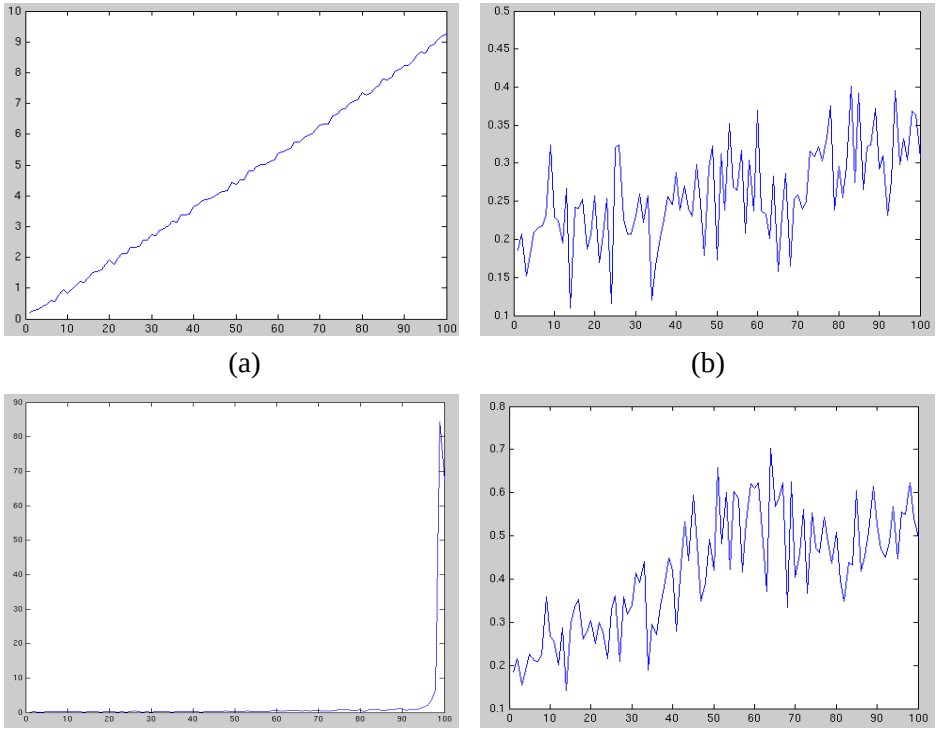

Figure 3: *(a) Reprojection error versus frame number of a simulated data set, using a 'cascaded' registration strategy. (b) Using a 'non-cascaded' registration strategy. (c) Feature points are lost as the frame increases. (d) Feature points are restored when the number falls below a certain cutoff.*

The idea here is that to register frame $\mathbf{F}_1$ to $\mathbf{F}_N$ we may replace the cascade of $N-1$ affine transforms with the single transform $\mathbf{A}_{1,N}$, estimated directly from the point sets $\mathbf{P}_1$ and $\mathbf{P}_N$ via the known point correspondences. Using the simulated data described above, the error as the number of frames also increases in a linear fashion, but its rate of growth is significantly reduced, as shown in Fig. 3b. This small increase in error per frame is due to the rounding error generated when iteratively applying the specified affine transform.

This 'non-cascaded' strategy however relies on the fact that all of the points have been successfully tracked throughout the frame sequence, which is usually not the case (for example, a proportion of the tracked points will leave the field of view as the camera pans). More realistically then, there is a point attrition that occurs during tracking that may be as high as a few per cent per frame. As the number of successfully tracked points declines through the sequence, the error in the registration model increases and eventually overtakes the error that would have occurred had the 'cascaded' registration strategy in Eq. (5) been used. A plot of this error is shown in Fig. 3c using the above simulation data. Here, the point attrition rate was set to two points each frame (*i.e.* one percent).

In summary, in order to minimise cascaded registration error, a 'non-cascaded' strategy can be used while there are sufficient tracked points to produce a good fit to the data. When there are insufficient successfully tracked points however, it is best to restart the registration process using a fresh set of feature points. In our implementation, we determine when to do this using a simple user defined cutoff value specifying a minimum acceptable number of feature points. Such a strategy results in the following formulation for the registration of frame $\mathbf{F}_N$:

$$\mathbf{F}_N = \mathbf{F}_1 \mathbf{A}_{1,k_1} \mathbf{A}_{k_1,k_2} \ldots \mathbf{A}_{k_m,N-1}.$$

Here, frame k_i designates a frame where the number of feature points successfully tracked fell below the specified level, and the tracking process was reset. It then continued to frame k_{i+1}, where again the tracking process was required to be reset, *etc.* Based on the above simulations, Fig. 3d shows a result using the combined 'non-cascaded' strategy with periodic resetting of the registration process. Here the cutoff value was set to 50 points.

2.3.3 Removing Outliers from the Set of Feature Points

In the feature tracking method, it is important to identify feature points that correspond to moving targets and remove them from consideration in the model fitting stage. This makes the approach more robust when, for example, there are large moving objects in the scene or when there is a significant amount of noise. On the other hand, failure to remove outliers can result in rather poor fitting of the registration models, which underpin the background modelling and subsequent motion extraction stages. The algorithm we use is called "RANSAC" (RANdom Sample And Consensus) [28], which is designed to fit models to data in the presence of a significant number of outliers. This algorithm is used widely in the computer vision community, in particular in the computation of scene homographies for constructing shape from motion. The algorithm could be applied to remove unwanted correspondences from other methods of registration, *e.g.*, optical flow vectors or tie-points. We would argue however that area-based methods that yield tie-points tend to smooth the motion due to moving targets into the estimates of the positions of tie-points. The RANSAC algorithm can be summarised as follows:

Objective

Robust fit of a model to a data set of feature correspondences S that contains outliers

Algorithm

- Randomly select a sample of s data points from S and instantiate the model (in our case, a parametric affine or projective transform) from this subset
- Determine the set of data points S_i which are within a distance threshold t of the model. The S_i is the consensus set of the sample and defines the inliers of S.
- If the size of S_i (the number of inliers) is greater than some threshold T, re-estimate the model using all the points in S_i and terminate
- If the size of S_i is less than T, select a new subset and repeat the above
- After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i

The feature set S is the set of feature correspondences between a given pair of frames in the sequence. We have found that in order to distinguish between background motion, foreground motion and the measurement error inherent in feature location, this pair of frames should be separated by at least 10 frames. Other values typical of our implementation are $1 < t < 10$, $T > 50\%$ of the number of matches, and $N \sim 2000$ iterations.

An example is shown in Fig. 4, where a frame from a video sequence taken from a ground-based handheld HDTV camera is illustrated (a subset of the frame is shown here for clarity). The KLT algorithm has been applied to track 500 features through the continuous sequence, which was subject to unconstrained camera motion and zooming. The white crosses are the inliers determined by the RANSAC algorithm and are used to estimate the registration model and the black crosses are outliers and are ignored. The feature points tracked on moving targets are deemed to be outliers and a number of points in the background of the image have been classified as outliers as well. This occurs because we have set the distance threshold t in the algorithm quite low so as to be sure to remove all the feature points that correspond to moving targets. We also require the point to be tracked through a certain number of frames (*e.g.*, 10) in order to provide a sufficiently wide baseline to distinguish between background motion, foreground motion and the measurement error inherent in feature location. Those points with insufficient track length are also deemed outliers. Generally there are ample inlier points remaining to fit a good registration model.

2.3.4 Error Correction

By analysing the reprojection error of the registration model, it is possible to determine automatically when the registration process has failed and to implement a recovery strategy. This is particularly important when generating mosaics of video imagery, as the process relies on an accurate registration through the entire video sequence. Figure 5 shows clockwise from top left a set of three consecutive frames from a video sequence of Mallala Raceway in South Australia. A mosaic of this scene is required but, as can be seen by the sudden jump between the second (top

Figure 4: *Removing outliers from the feature set. White crosses indicate inliers used for image registration and black crosses indicate outliers.*

right) and third (bottom right) frames, the signal received at the ground station has momentarily dropped out. Typically, off-the-shelf mosaicing packages will break down in such cases because they require smooth continuity through the sequence of frames. The feature points that have been tracked have also been lost over this region and therefore our estimates of the registration model will be inaccurate. This is directly reflected in the reprojection error, as defined in Eq.(1): The bottom left of the figure shows a plot of this error and reveals a pronounced increase in error when there is a loss of signal. It is a simple matter to implement a threshold (at, say, a value of 5) to automatically determine when such errors occur.

We have subsequently implemented a simple error recovery strategy that will omit a block of corrupted frames from the video sequence and register the two frames at either end of the block, to form a continuous mosaic. The process takes the point sets from these two frames and tries to find the optimal translation between the two disparate point sets with no *a priori* correspondence information, using a brute force search of order N^2. The essential idea of the algorithm is to use a pattern matching approach that translates one point set to the other, through all possible translations, establishing correspondences through a simple distance threshold. The process relies on at least some spatial overlap between the two points. For point sets that have of the order of hundreds of points, the implementation time of the algorithm is acceptable. An alternative approach reported in [10] can find the optimal affine transform between two point sets directly, but it assumes *e.g.*, that the point sets overlap spatially (a generalisation of the algorithm is in progress).

Objective

Find optimal translation between the two point sets $\mathbf{P}_i, i = 1 \ldots N$ and $\mathbf{P}'_j, j = 1 \ldots M$, with no correspondence information.

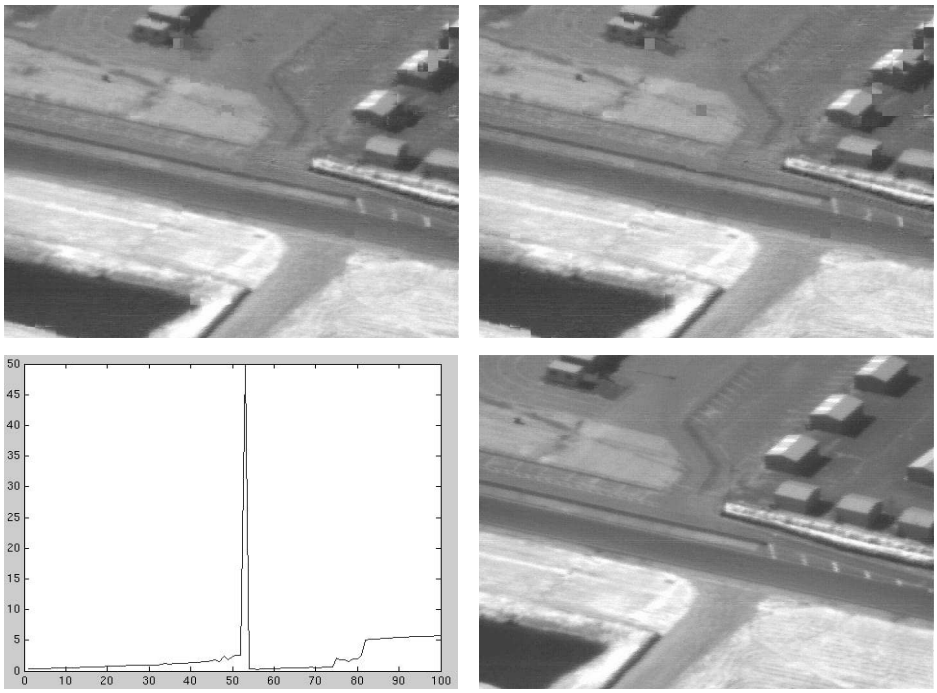

Figure 5: *Clockwise from top left: Three consecutive frames from a sequence with drop out and other errors in signal transmission. Bottom left: The reprojection error can be used to automatically predict when registration has failed.*

Algorithm

For all $i = 1 \ldots N$, $j = 1 \ldots M$

- Form a translation set $\mathbf{P}'' = \mathbf{P} + \mathbf{t}_{i,j}$, where $\mathbf{t}_{i,j} = \mathbf{P}'_j - \mathbf{P}_i$
- Determine set of correspondences between $\mathbf{P}''$ and $\mathbf{P}'$: point $\mathbf{P}''_n$ corresponds to point $\mathbf{P}'_m$ if the distance $\|\mathbf{P}''_n - \mathbf{P}'_m\| < t$
- Record mean distance $\bar{d} = \frac{1}{K} \sum_{k=1}^{K} \|\mathbf{P}''_{n_k} - \mathbf{P}'_{m_k}\|$, where K is the number of correspondences

The optimal $\mathbf{t}_{i,j}$ is the one that gives the lowest mean distance $\bar{d}$ over all i, j. Given the optimal translation $\mathbf{t}_{i,j}$, a solution for the parametric transform between the sets $\mathbf{P}'$ and $\mathbf{P}''$ may be found by solving $\mathbf{P}' = \mathbf{P}''\mathbf{A}$ using a least squares approach.

The result applied to the example in Fig. 5 is shown in Fig. 6. Here the mosaic has been formed successfully with no apparent error in the result. We plan to investigate methods to incorporate available meta data such as geocoding information to improve the mosaic result. In particular, our collections include highly accurate positional information that describes camera location and pose and this information may be incorporated into the solution through the use of techniques such as bundle adjustment [51].

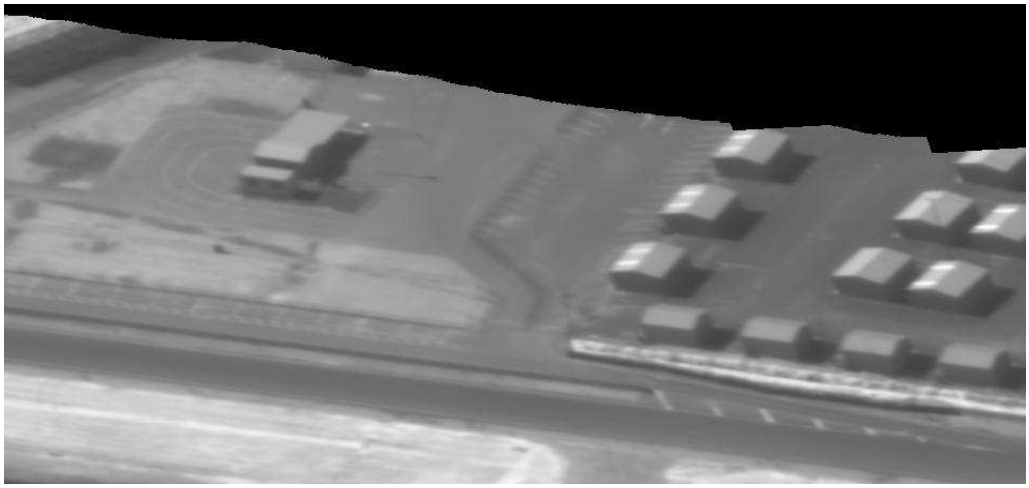

Figure 6: *Resulting mosaic in region of signal drop out, after error correction has been applied.*

2.3.5 Tracking in Relatively Featureless Regions

One of the drawbacks often cited with regard to feature tracking is that it is generally not applicable to video of scenes that are featureless. However, we have yet to find a real example where this has been the case in our airborne and ground-based sensor collections (other domains of imagery might pose a problem however, for example tracking and recognising features in imagery of faces, which have large regions of smooth texture). For example, in the interlaced standard definition frame shown in Figure 7, the subject is a relatively featureless desert terrain but there is still sufficient features in the scene to find and track over the entire scene (as shown in black and white at the bottom of the figure, and discussed further below). This particular sequence consisted of 29 non-consecutive frames taken from a moving airborne platform, where images could be two to four frames apart. We found that the algorithm tracked features that could shift by as much as 15 pixels between frames. The camera motion in the sequence is parallel to the vehicle motion and there is significant vertical furrowing evident in the imagery. This imagery posed some problems for optic-flow based techniques, *e.g.*, with interlacing there is more "energy" in the vertical direction and this tends to erroneously weight the registration in the vertical direction.

2.3.6 Registration Performance Summary.

We have investigated two registration techniques with regard to VMTI, one region-based and one feature-based. The approach used in the prototype VMTI system with some success was region-based. This hierarchical, correlation-based technique, was outlined by Privett and Kent [45]. Our attention has now shifted to the feature-based approach described above and by Jones *et al.* [36]. Both algorithms have the capability to model the image-to-image transformation with a complexity up to projective.

Moving target detection in video imagery usually operates on two frames separated by a short time interval, during which the camera motion is relatively uncomplicated; an affine transformation model is usually more than sufficient. Indeed there are strong arguments in favour of keeping

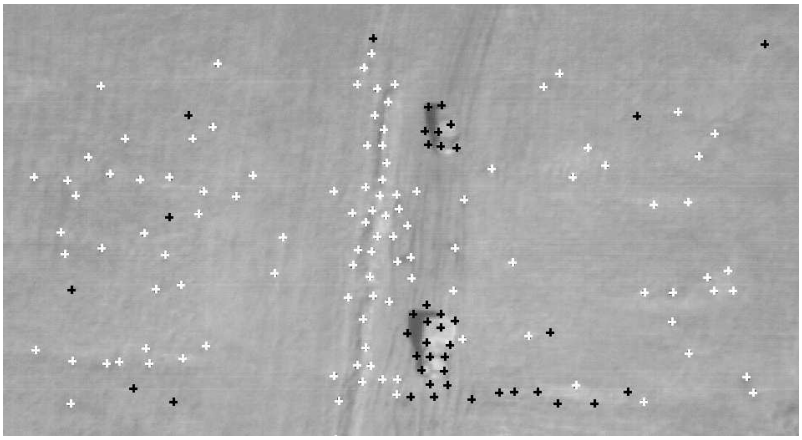

Figure 7: *Example frame from a sequence that is relatively featureless. Features are shown after classification with the RANSAC algorithm, where black points indicate outliers.*

the degrees of freedom low to prevent overfitting, particularly if the scene contains unevenly distributed 3-D structure.

Background modelling requires frames to be registered which are separated by longer time intervals during which time the potential exists for more complex platform and camera maneuvers to have taken place. This presents several problems. Firstly the frame to frame mappings may need to be more complex *i.e.* projective. These additional degrees of freedom offer the potential for overfitting or for misregistration by convergence to a local minima. In a similar vein, the range of durations between pairs of frames means that, in the case of our region-based approach at least, the optimisation schedule for the algorithm must be comparatively loose so as to encompass the more highly separated pairs. At best this will impact on execution time, and at worst it may impact on the precision of the convergence or may cause convergence to implausible minima.

3-D structure presents serious problems. Buildings, although usually being weakly textured, are generally strong exhibitors of the types of features that are often integral to the registration fit metric (i.e. lines and corners).

The region-based approach considers the correlation between the two whole images. Given imagery containing a textured ground-plane and some 3D objects it will reach a compromise registration (that could be viewed as an average weighted by image local structure density). Our feature based approach is in some sense more robust. While the proportion of features emanating from the 3-D structure is low, it should have little or no impact. However, because the 3-D structure is high in strong corners, a comparatively low proportion of the imagery being populated by structures such as buildings may cause alignment to building roofs rather than the ground, particularly if the buildings are of uniform height *i.e.* the relative displacement between corresponding corners is uniform.

These expectations have been borne out by results. We'd expect the region-based approach to recover from translation of 10-20 % of the image width and 10-20 degrees of rotation. Performance is reduced when the images exhibit large perspective variation (change in elevation angle). The feature based approach has been applied to a large and varied set of imagery and has performed well, even when the imagery has exhibited relatively weak and sparsely distributed features. In the presence of 3-D structure, the region-based approach solution tends to drift around locally

from frame to frame, and there is a gradual increase in registration error as the platform moves further away from the target image. Under the same conditions, the feature based approach gives precise alignment of building roofs resulting in more pronounced misregistration of the ground plane features.

2.4 Background Modelling

This section considers the construction of a background model using those anomalies in individual video frames that can be identified. One advantage of this approach over image differencing is that, usually, the whole silhouette of the anomalous object is made apparent rather than (predominantly) changes in occlusion and disocclusion *i.e.* two incomplete object segmentations or, in the extreme cases, two complete segmentations per target (and a resulting association problem). In addition, the use of a background model means that an object introduced to a scene (such as a briefcase) will be persistently visible even if static.

The main considerations with regard to constructing the background model are: choice of background function, choice of colour space, the methods used for bootstrapping and maintaining the model, and the decision mechanism for identifying statistical outliers. The most appropriate choice depends on the scene content and how it changes over time, imaging conditions, *etc.* As a result, many options are available, and ADSS can be configured accordingly.

2.4.1 Background Functions and Maintenance

Background models are usually constructed from a number of frames. The colour distribution of each pixel over time could be modelled by treating each colour component independently or by adopting a multivariate route. We consider both. The assumption is that the camera is static and, therefore, that over time, a pixel images the same area of the scene repeatedly. As we have seen, some camera motion can be removed by registration, however, the background model should be capable of capturing lesser movements resulting from camera shake.

The background is a dynamic system, and its model must be regularly recomputed or updated accordingly. This is known as background maintenance. The sophistication of these update techniques varies enormously, from straightforward iterative update rules to techniques that consider the motion histories of component objects. Depending on the duration of the observation, these updates may need to accommodate gradual illumination changes (*e.g.*, sun precession during the day), sudden illumination changes (*e.g.*, electric light switches and cloud/sun transitions) and the introduction of shadows cast by objects within and outside the scene. They must also accommodate changes in the stability of the camera (*i.e.* shake) and persistent scene movement such as moving vegetation. And at some point, it would be desirable that objects introduced to the scene that present no interest should become part of the background model.

Overview

The Gaussian background model is reasonable for characterising the pixel variation in a static scene, however, the mean and variance will be biased by statistical outliers, including moving objects. We characterised the distribution using robust statistics *i.e.* via the median and the median

absolute deviation from the median, respectively. This provided a more satisfactory model in the presence of outliers when dealing with small numbers of frames, and in particular, the system can be bootstrapped without a need for the scene to be evacuated of moving objects. Given the background model,(μ,σ), outliers can be identified by exceeding some user-specified multiple of standard deviations from the mean.

A Gaussian model can be made to adapt to slowly changing illumination conditions by recursively updating the model using a simple adaptive filter such as (6) and (7) below. Koller [37] and others have used Kalman filters for the adaptation. However, in outdoor environments, windy conditions may precipitate camera shake and vegetation movement, both of which may result in several object classes being in view over an extended period of time. Each of these should be modelled individually. The most common approach is to employ a mixture of Gaussian model [26, 27, 24, 39, 41, 40]. Grimson *et al.* used between 3 and 5 Gaussian distributions - notionally one per background class. Similarly, Friedman and Russell [24] used a three Gaussian model (as have we) to represent road, shadow and vehicle distributions.

The mixture is weighted by the frequency that each Gaussian explains the background. The mathematical model is given by

$$\{X_1, ..., X_t\} = \{I(x_0, y_0, i) : 1 \leqslant i \leqslant t\}$$

$$P(X_t) = \sum_{i=1}^{k} w_{i,t}\eta(X_t, \mu_{i,t}, \Sigma_{i,t}),$$

where $w_{i,t}$ are mixture weights, and

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} \mid \Sigma \mid^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T\Sigma^{-1}(X_t-\mu_t)}.$$

And we take

$$\Sigma_{k,t} = \sigma_k^2\mathbf{I}$$

for ease and rapidity of inversion. A small offset is added to the variances to prevent division by zero when no image noise is present. This might result from local image saturation or from other imaging peculiarities. The mixture of Gaussians can be derived using an Expectation Maximisation (EM) clustering algorithm, variations of which are numerous. In our experiments we evaluated an unsupervised EM algorithm proposed by Figueiredo and Jain [23] together with a k-means algorithm [57]. The former selects the number of component distributions and does not require careful initialisation. We have used this approach very successfully for partitioning target segmentations into spatio-colour clusters, however, in this endeavour, the data was sufficiently at variance with the ideal (due to noise, pixel drop out etc) that the author experienced serious reliability issues. At present we favour the k-means approach. This is an iterative process based on the distances of samples to current estimates of cluster centres. Unlike the EM algorithm which generates overlapping Gaussian distributions, k-means partitions the samples neatly into nonoverlapping labelled clusters [11]. This is appealing because it offers some potential to map the cluster back to the source land-use or object in the scene. K-means requires the desired number of clusters be specified, however, in non-ideal data it is still possible that a particular pixel may exhibit a lesser number of clusters. This must be recognised in the resulting output before an attempt is

made to detect anomalies. The k-means approach suits a 2D, orthogonal colour space well (Section 2.4.2). Inputs to the k-mean clusterer may be standardised to prevent spurious domination of one channel.

Background maintenance for the Gaussian Mixture model was done using the following update rule [55] once the corresponding distribution had been identified

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha(M_{k,t}),$$

where α is the learning rate and $M_{k,t}$ is 1 for the matched model but otherwise 0, and

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t, \tag{6}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t-\mu_t)^T(X_t-\mu_t). \tag{7}$$

And a second learning rate, ρ, is given by

$$\rho = \alpha\eta(X_t \mid \mu_k, \sigma_k).$$

If X_t does not belong to any of the K distributions (to within $n\sigma$) then a new distribution is created with a corresponding mean, high variance and low weight.

These approaches can adapt to gradual and more rapid changes but sudden switches in lighting changes still present serious problems. Proposed solutions include the use of hidden Markov models. These have been used to model three state environments: foreground, background and shadow [49]. Edge features have also been used to model background [61, 34] in an effort to achieve illumination invariance.

An Airborne Focus

In the case of an airborne moving camera however, the extent of the temporal window is restricted and so the contribution made by moving objects to the formation of the background model is relatively high. Background modelling based on standard Gaussian statistics tends to include this contribution into the background and thus diminish the effectiveness of VMTI. We have found that robust statistics, such as the median filter and maximum absolute deviation, yield superior results and are the preferred method of background modelling in the case of the moving sensor. Figures 9 and 10 (on the following pages) illustrate the benefit of using robust statistics by the relative impact on background variance caused by a person moving through the scene (at low sampling rate).

More formally, given a stack of N registered images $\mathbf{F}_i$, the background model $\mathbf{B}$ is given by

$$\mathbf{B}(x,y) = \text{median}\ \{\mathbf{F}_i(x,y),\ \ i = 1 \ldots N\}. \tag{8}$$

An important consideration is that, although the images $\mathbf{F}_i$ are registered to one another, they do not typically fully overlap (in fact they might overlap by as little as fifty percent). The total combined area of the stack of images $\mathbf{F}_i$ is inevitably larger than that of any single frame. In order to use as much information as possible to construct the background model, and to avoid unwanted boundary artifacts, we endeavour to construct $\mathbf{B}$ over the entire combined area. However, a given coordinate (x,y) in $\mathbf{B}$ may not be contained by all N frames, in particular at coordinates on the

Figure 8: *Mean background model in red, green, blue colour space.*

periphery of $\mathbf{B}$. For such coordinates, we can only compute the median filter over those frames that contain that coordinate. If the number of such frames falls below some acceptable level, say five, we do not compute a background model for that coordinate at all, as the statistic become unreliable.

An accepted robust statistic for variance that can be used in conjunction with the median statistic is the median absolute deviation, or MAD, defined by

$$\text{MAD}(x,y) = \text{median } \{\|\mathbf{F}_i(x,y) - \mathbf{B}(x,y)\|, \;\; i = 1 \ldots N\}, \tag{9}$$

where $\mathbf{B}(x,y)$ is the background model as defined in Eq. (8). It is usual to make MAD consistent with a normal distribution by dividing by $\Phi^{-1}(\frac{3}{4})$, roughly 0.6745 [30]. Again, we can only compute the MAD over those frames that contain the given coordinate.

2.4.2 Colour Model

All commonly used colour transformations are available in ADSS. In this application RGB was not pursued due to the high degree of correlation between the red, green and blue channels, and the dominance of intensity (Figure 8). Essentially, the discriminatory potential of the colour information is not fully realised, and the intensity and colour components cannot be assessed independently.

HSV (Hue, Saturation, Value) does have the desirable property of separating intensity from hue (Figure 11), one benefit of which is the ability to use a simple heuristic to distinguish between object movement and shadow change. However, the continuous, circular nature of the hue feature results in a discontinuity (seen here in the sky (Figure 11)) which results in a region of high variance when changing scene and imaging conditions are considered (Figure 12).

It seems that this characteristic of HSV space is usually ignored by the community, and indeed this may be reasonable depending on the operations being applied to the hue values. Some of our discomfort with the hue feature was soothed by the following approximations which were used successfully in some of our experiments. The angular mean is given by

$$(\cos(\theta), \sin(\theta)) = \left(\frac{\frac{1}{n}\sum_i \cos(\theta_i), \frac{1}{n}\sum_i \sin(\theta_i)}{r} \right)$$

Figure 9: *Background variance in red, green, blue colour space.*

Figure 10: *Robust estimate of variance in red, green blue colour space.*

Figure 11: *Mean background model in HSV (Hue, Saturation, Value) colour space.*

Figure 12: *Background variance in HSV (Hue, Saturation, Value) colour space.*

Figure 13: *Mean background model in Principal Component Space (PC1, PC2, PC3) colour space.*

where

$$r^2 = \left(\frac{\sum_i \cos(\theta_i)}{n}\right)^2 + \left(\frac{\sum_i \sin(\theta_i)}{n}\right)^2$$

and the angular variance is given by $2(1-r)$ [58].

A principal component based feature space has also been considered (Figure 13) based on the work of Ohta [43] and Dodd [21]. They noted that images belonging to a common class, such as natural outdoor scenes, have very similar principal components. These are obtained by diagonalising the covariance matrix of the RGB feature vector, and can be approximated by a single set of transformations that can be applied to any image in the class. It should be noted, however, that as a result of making this generalisation, the transformed components will contain some correlation.

With regard to the set of outdoor images that we examined, in all cases PC_1 is approximately $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t$, PC_2 is dominated by either $(-\frac{1}{2}, 0, \frac{1}{2})^t$ or $(\frac{1}{2}, 0, -\frac{1}{2})^t$, and PC_3 by $(-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4})^t$. These findings are in agreement with those of Ohta, who then proceeded to derive a set of features f_1, f_2, and f_3 for representing colour information.

$$f_1 = \frac{R+G+B}{3}$$

$$f_2 = \frac{R-B}{2} \quad \text{or} \quad f_2 = \frac{B-R}{2}$$

$$f_3 = \frac{2G-R-B}{4}$$

To determine the information held by a particular PC image, it can be replaced by its mean, and the PCA transformation inverted to produce a modified RGB image for comparison with the original. Clearly, f_1 is the same as the Value feature in HSV. The second PCA is a colour feature, and the third component captures any residual information, that is, bright colours and some texture.

In his work on segmenting images of natural outdoor scenes, Ohta [43] compared the performance of his feature set with seven others in common use. He found very little difference, and considered it due to the almost 2 dimensional nature of colour information, *i.e.* intensity and one chromatic feature. Land's psychophysical experiments of the 1950s drew similar conclusions

[38]. He demonstrated that perceptual colour is almost two dimensional, and showed that the perception of colour is not purely a local phenomenon but is dependent on the spatial content of the scene.

Since we regard perceptual colour as being almost two dimensional, intensity and a colour component, we usually consider the third component to be noise and discard it. A two dimensional, orthogonal colour space is very appealing both for its ease of manipulation and for the computational speed that it promotes.

Summary

With regard to the median-MAD background model, much of the airborne imagery processed to date has been treated as monochrome imagery, that is, it is left as is, or if necessary an intensity feature has been derived *i.e.* $I = (R+G+B)/3$. In some instances, the mapping $0.3R+0.11G+0.59B$ has been used to mimic the response of the human visual system.

ADSS supports most colour spaces in common use. Some of these, principally the HSV and PC spaces, have been applied to ground-based imagery. In these cases, each channel has been treated independently and the results OR'd, that is, if an anomaly is found in any of the channels then motion is assumed.

2.5 Anomaly Detection

Given a background model in the form of robust estimates of mean background (**B**) and its standard deviation, it is trivial to compute the probability that a particular pixel is consistent with the background model (in terms of standard deviations). Outliers can be identified using a statistically meaningful multiple, n, of the standard deviation.

$$\mathbf{M}_i(x,y) = \frac{\mathbf{F}_i(x,y) - \mathbf{B}(x,y)}{\text{MAD}(x,y)/\Phi^{-1}(\frac{3}{4})} > n. \tag{10}$$

Here $\mathbf{M}_i$ is a binary motion image corresponding to frame $\mathbf{F}_i$ and (x,y) is the set of coordinates in frame $\mathbf{F}_i$.

As the image $\mathbf{M}_i$ is in the registered frame of reference of $\mathbf{F}_i$, it is transformed back to the frame of reference of the original unregistered video frame corresponding to $\mathbf{F}_i$. This is a simple matter of applying the inverse of the transform that was used to generate $\mathbf{F}_i$. The resulting stack of motion images may then be fed into the tracking stage, *e.g.*, a particle filter [48] or a simple heuristic tracking algorithm (such as that described in the following section). It should be pointed out that, because it is based on frame differencing, the motion estimation $\mathbf{M}_i$ does not produce strong evidence in favour of a moving target if, by chance, it occupies a similar position in colour space as the background. The result may be incomplete moving objects or objects that are missed altogether. We suggest that using a target/foreground model could alleviate the fragmentation problem. The impact of registration errors and occlusion changes on the background model, and consequently on detection performance, should also be borne in mind. As the sensor moves, regions on the ground occluded by 3D structures will gradually change. The impact is difficult to predict as it depends on image content. However, relative movement of image structure (whether on the ground or elevated from it) will lead to false alarms, and as the grey level variance estimate

close to a structure is likely to be raised, changes that take place close to such region boundaries are more likely to remain undetected. This places additional heavy constraints on the size of the temporal window.

2.6 Heuristic Tracking Algorithm

As the final part of the VMTI implementation in ADSS, a simple but effective heuristical tracking algorithm was implemented to take the output of the VMTI process and produce ADSS detection messages for moving targets. Other tracking algorithms, such as particle filters and probabilistic multi-hypothesis tracking (PMHT) are currently under development in ADSS and will be the subject of a future report. The heuristic algorithm works on the concept of temporal integration, or the accumulation of objects in the same location over a period of time. The longer the object is sustained through the time in the VMTI sequence, the more likely it constitutes a true moving target. In contrast, noise in the motion estimation, *e.g.*, caused by flickering light or changes in perspective, is not typically sustained over a long period of time and can be filtered out by the temporal integration process.

To begin with, a user specified threshold t is applied to the motion estimation in order to produce binary images which may then be labelled for further analysis,

$$\mathbf{M}_i(x,y) = \frac{\mathbf{F}_i(x,y) - \mathbf{B}(x,y)}{\text{MAD}(x,y)/\Phi^{-1}(\frac{3}{4})} > t.$$

Here the threshold t typically has a value ranging from 2 to 3 and essentially specifies the number of deviations from the expected background value that a pixel must have to be considered part of a moving target. The application of a threshold tends to leave only those parts of the VMTI imagery for which there is strong evidence of moving objects. Typically however, a certain amount of specular noise is also passed by the thresholding process. The stack of images $\mathbf{M}_i$ may then be labelled independently and objects thresholded on the basis of user specified thresholds applied to simple shape attributes such as area and/or size of bounding box. This tends to remove a lot of small impulse noise, and leaves larger objects caused by changes in illumination, movement of trees etc. The connectivity of an object through time is then established by iteratively tracking it through the sequence, using overlapping shape attributes. More specifically, object $\mathbf{A}$ in one frame is deemed to be connected to object $\mathbf{B}$ in the next if its centroid lies within the bounding box of $\mathbf{B}$. A path length can be assigned to each object, given by the number of frames that it can be tracked through the sequence. Objects may then be filtered on the basis of path length using a user specified threshold.

An alternative method to 2D labelling followed by connectivity analysis is to use full 3D labelling of the stack of thresholded $\mathbf{M}_i$ images, which automatically establishes connectivity in the temporal dimension. Temporal integration is then realised as the computation of volume statistics of the binary objects, which may be thresholded leaving only high volume objects that are (typically) moving objects sustained over longer periods of time. However, the labelling of 3D objects is relatively labour intensive and requires significantly more memory resources. Moreover, a 3D labelling algorithm has yet to be implemented in the ADSS.

3 Examples

In this section, we present some results produced by the VMTI subsystem on a range of imagery that highlight some of the issues that arise when implementing VMTI for a moving sensor. A relatively straightforward example of VMTI is shown in Fig. 14. At the top left of the figure is an example frame from a video sequence taken from an airborne MX20 sensor. A car is moving along the road and the camera is following the car through the sequence. It is standard interlaced video of 24 frames per second, where each frame is of size 704×480 pixels. One of the issues with using interlaced video is that it can lead to artifacts like those shown in the top right of the figure. Here the camera has panned suddenly and this results in a splitting of the separate components of the interlaced frame. This effect will cause many registration algorithms to fail. For example, optical flow based techniques will tend to find more "energy" in the vertical direction and erroneously weight the registration in the vertical direction. A de-interlacing algorithm (*e.g.*, a simple sampling in the vertical direction) needs to be employed if such algorithms are to be used with any reliability. The KLT registration algorithm however was very robust to this problem and good registration results were obtained without the application of a de-interlacing algorithm. As shown by the result at the bottom left of the figure, the VMTI results for such frames can exhibit very high noise caused by the apparent difference between the frame and the model of the background. Although erroneous moving targets may well be found in this frame, the application of the temporal integration can remove this noise, as shown by the result for this frame at the bottom right of the figure.

Another example, using a video from a handheld HDTV camera, is shown in Fig. 15. The frame rate for HDTV video is typically 60 frames per second, where each frame is 1280×720 pixels in size. The camera underwent unconstrained panning and zooming while following the moving subject, as can seen by comparing the two frames at the top of Fig. 15. Unconstrained zooming can pose significant problems for registration algorithms that are sensitive to scaling. Although our feature tracking method is in fact sensitive to scaling, and features are lost during zooming, they are immediate replaced by new ones. Overall then, the approach is robust to zooming. The sequence also contains significant movement from *e.g.*, leaves on the trees and bushes, occasional cars moving through the scene, workmen in the background of the scene and flickering sunlight off cars and windscreens. All these examples of motion in the scene were successfully discounted from the video registration process by applying the RANSAC algorithm to the feature points that were tracked, as discussed in Section 2.3.3. This provided a very solid background stabilisation from which reliable VMTI results could be obtained, as shown by the examples at the bottom of the figure. Here the subject has been successfully detected as a moving target, although the subject is incomplete in parts because there is insufficient discrimination between the intensity of the subject's clothes and the intensity of the background model that has been constructed. If it is important to accurately delineate the boundary of the moving object, a further segmentation process, such as a snake or watershed algorithm, could be applied.

The VMTI process extracts all objects in the scene that are deemed to be moving, and this includes objects such as moving leaves on trees and fluctuating light patterns, which are generally undesired for the purposes of VMTI. In our approach, this apparent motion can be removed during the temporal integration phase. More sophisticated techniques may also be applied to discount objects on the basis of their movement patterns, *e.g.*, to remove objects that do not have a consistent direction of movement. The alternative approach, which has been widely adopted for the case of the stationary sensor, is to use a background model based on multimodal Gaussian distributions.

Figure 14: *Example VMTI analysis on an interlaced video sequence taken with an MX20 camera. Top left: Sample input frame. Top right: Close up of interlacing striping caused by sudden camera panning. Bottom left: Output image from VMTI process, showing excessive noise from interlacing artifacts. Bottom right: Result after tracking algorithm has been applied; the noise has been removed by temporal integration.*

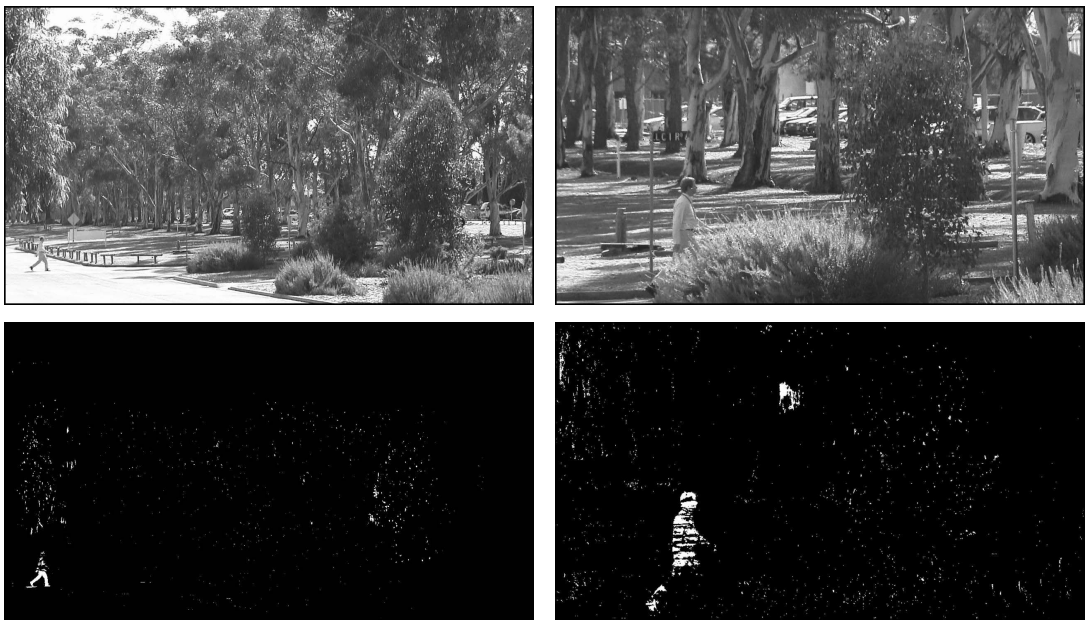

Figure 15: *Example VMTI analysis on an non-interlaced video sequence taken with a handheld HDTV camera. Top: Two sample frames showing unconstrained panning and zooming. Bottom: Corresponding VMTI imagery showing moving objects.*

This has the ability to automatically absorb periodic movement such as moving leaves and flickering light into the background model, through the use of its multiple modes, so that such motion is not exposed in the frame differencing stage. As discussed above however, in the case of the moving camera the extent of the temporal window used is very restricted and so the contribution made by moving objects to the formation of the background model is relatively high. Background modelling based on Gaussian statistics tends to average this contribution into the background and thus severely diminish the effectiveness of VMTI. We have found that the robust statistics we use, such as the median and MAD statistic, yield superior results. It could also be argued that the classification and discrimination of different types of independent motion in the scene might conceptually best be handled separately from the background formation stage. This allows a deeper analysis of an object's motion, *e.g.*, by applying shape statistics and patterns of behaviour rather than just pixel intensity distributions, and thus potentially more reliable discrimination.

Another example of VMTI on HDTV data is shown in Fig. 16, illustrating frames from a video sequence taken of Parafield airport. The sequence is particularly challenging due to the high density of structures which can induce errors in registration and motion estimation, due to the change in perspective of the structures as the camera moves through the scene. This means that the temporal window used for VMTI should be relatively short so as to minimise the effects of change in perspective. On the other hand, there is a wide array of moving targets in the scene of different size, ranging from people, cars, trucks and planes. As many of these objects are moving relatively slowly, the temporal window should be relatively long so as to minimise the contribution made by these moving targets to the background model. In this example, it is not possible to select the temporal window size to fully satisfy both constraints, and this leads to two types of errors appearing in the VMTI results: erroneous moving targets due to perspective errors and incomplete or missing moving targets due to insufficient background discrimination.

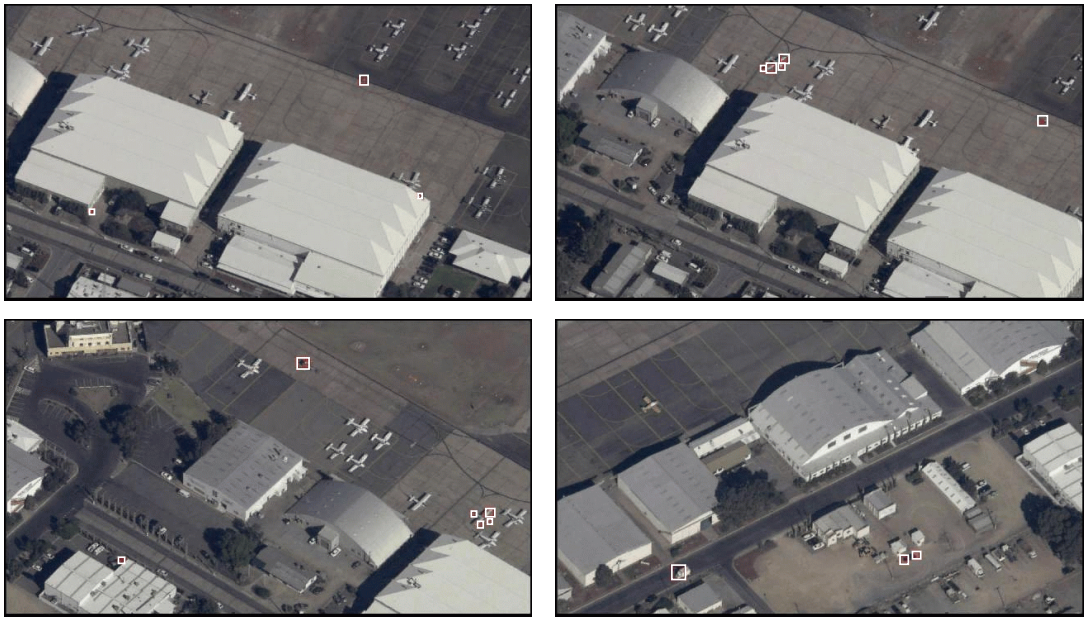

Figure 16: *HDTV sequence of Parafield airport, showing four examples of VMTI results. See text for details.*

In the top left frame of the figure, three objects have been detected. Only the top-most object is a true moving target (a person walking across the tarmac); the other two targets are caused by perspective errors. The top right frame again shows moving targets; this time a slow moving plane has been detected but only its leading edge was found. This is because it is moving too slowly for the selected temporal window size, and much of its shape (its wings and tail portion) has therefore been partially absorbed into the background model. The bottom left frame shows a similar situation, where this time it is the trailing edge of the plane that has been detected, along with the cab of a moving truck. The final frame at the bottom right shows a relatively fast moving van being detected, along with the leading and trailing edges of a slow moving car.

Perhaps one of the most difficult VMTI examples we have encountered is shown in Fig. 17. Here the sequence is from airborne video surveillance taken with an MX20 sensor. The targets to be detected from the VMTI process are very small; the aim is to detect targets that would normally be overlooked by the analyst. This example was used for the comparative analysis carried out and reported on in Section 4. The top frame in the figure shows two such targets (cars on roads) circled in white. The result from the VMTI process is shown in the bottom frame of the figure. Here the left most target, the dark car travelling down the road has been readily picked up by the VMTI process. The central targets however are rather poorly resolved. Moreover, there is a significant amount of noise in the VMTI results, due to misregistration errors in this cluttered environment. This noise is much more intense in other parts of the sequence, when the camera pans or clouds move through the sequence. Although these errors are fairly small in size, they are typically of the order of the same size as the targets that are sought. Our standard heuristic tracking process, based on shape analysis and temporal integration, has proven insufficient to reliably track the central targets in the sequence. However, more sophisticated tracking techniques, based around Kalman filtering and currently available from colleagues at DSTO [48], are able to use the output from our VMTI process to track the targets through a significant amount of noise. We are currently in

the process of implementing such algorithms in ADSS, and the augmented approach will be the subject of a future report.

As a final example to illustrate the versatility of our VMTI approach, Fig. 18 shows a pair of satellite images taken of a mining site in Australia at different times of the year. A typical problem faced by the image analyst is to register such images together for the purposes of detecting change between the imagery. We are able to cast this into a VMTI problem by simply defining the pair of images as two consecutive frames in a video sequence. Our standard VMTI process of feature detection, outlier removal, image registration and frame differencing may then be applied as is, and the results from the VMTI process are the required change detection results (note that in this case it is not necessary to construct a background model). The process works very well when there is sufficient overlap between the images for the KLT feature tracker to track features from one frame to the next. Typically the displacement between corresponding features must be less than about 10 pixels. The process will not work well however when there is significant rotation and/or scaling between the image frames.

4 Comparative Analysis

In this section we present the results of an evaluation carried out between our VMTI system, which we will refer to as ADSS VMTI, and two other VMTI systems. The first, "Industry VMTI", is the result of a short-term contract of several months duration carried out by a software company for DSTO, to deliver a real time VMTI system based on their existing intellectual property. This was essentially a no-frills Windows application that demonstrated the company's capacity to implement real-time VMTI. The second, "ARIA" [12], is a VMTI system developed within the MATLAB programming environment by Dr. Robert Caprari, ISRD, over a period of approximately 9 months and completed in March 2004. It currently operates at approximately 400 times slower than real time, and for that reason it is not considered to be a viable system for deployment in its current form. It is hoped that in the near future the ARIA and ADSS VMTI systems can be combined after ARIA is ported into ADSS, whereby the strengths of both systems can be exploited.

4.1 Experiment and Results

For the purposes of comparing the three systems, a five-minute video sequence was chosen and ground truth information extracted. The sequence was of standard definition (704 by 480, 30 frames/sec) IR airborne video surveillance from the ISR Testbed collection and is the "difficult example" that was illustrated in the previous section (in Fig. 17). The sequence recorded the movements of a convoy of 3 targets as they undergo maneuvers through the hills of Woodside, SA, and includes occlusions, fast camera pans, and footage of fast moving traffic on a nearby freeway. The ground truth consisted of a count of the number of actual moving targets present in each frame in the sequence, as judged by an image analyst. When moving targets became occluded or came to a standstill in certain frames, they were not judged to be moving in those frames. The results for moving targets for each system were then compared to the ground truth, by noting for each frame the number of targets successfully located and the number of false targets located. The false alarm rate (FAR) for the sequence was calculated by dividing the total number of false alarms recorded

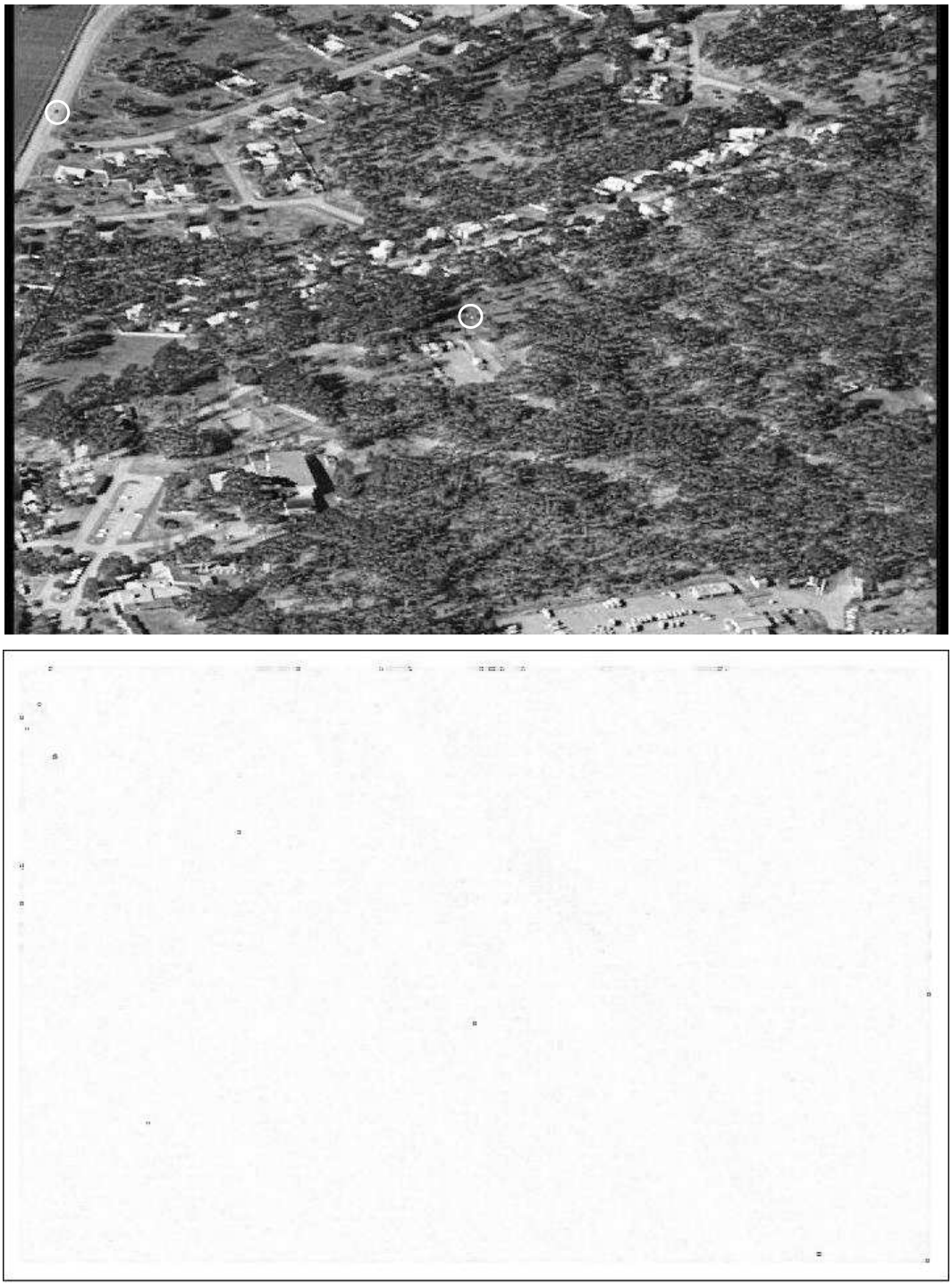

Figure 17: *Detecting small targets in airborne surveillance. Top: frame from input sequence. Bottom: Corresponding VMTI frame.*

Figure 18: *Applying VMTI process to change detection in satellite imagery. Left and Middle: Input images. Right: Difference image after registration.*

by the number of frames in the sequence. The probability of detection (PD) was calculated as the total number of targets successfully detected divided by the total number of targets identified in the ground truth. The results for the three systems are shown in Table 1.

	Total FAR	FAR	PD	Frames/s	Report/s
Industry VMTI	446	0.0496	0.4285	30	8.37
ARIA	0	0.0000	0.1913	0.075	30
ADSS VMTI	47	0.0053	0.5963	27	30

Table 1: FAR, PD and timing results for the three VMTI systems

In the interests of drawing a fair comparison between the three systems, certain concessions were made when calculating these statistics. These are detailed in the following section. Broadly speaking however, we may draw the following conclusions:

- Industry VMTI provides a very fast VMTI system on relatively cheap hardware. However, the higher processing speed appears to have come at the cost of the FAR and PD rates. In particular the current FAR would not be acceptable for a useful deployed system.
- ARIA produces an impressive FAR rate of zero (more generally, it has been recorded to produce one false alarm in an hour of video footage), and an acceptable PD as the analyst only has be to be alerted once for a given moving target through the sequence, before investigating further. However, its processing time is currently too slow to be acceptable.
- The ADSS VMTI produces the best PD in an acceptable processing time, though lag may be an issue (see below). Further work on the FAR rate may be necessary (it found one false alarm in the five minute sequence), probably by tuning the parameters to allow a lower PD.

4.2 Analysis

The Industry VMTI System

The focus of the Industry VMTI system was to deliver as good a VMTI system as possible while maintaining a real time implementation, and this has come at the expense of a higher FAR

and lower PD. The system has demonstrated that it is certainly possible to do real time VMTI, and we are not aware of any other COTS systems capable of real time VMTI. (For example, the Jam system, developed by Pyramid Vision Technology, requires hardware acceleration for real time tracking, and moreover the tracking requires initialisation by the user.) As detailed below, the ADSS VMTI system is currently capable of near real time implementation.

The system currently has a bug in that there is no synchronisation between the input frames and the tracked objects reported for each frame (due to a bug in the third party MPEG decoder). It was therefore necessary to synchronise the reporting with the input frames by hand, and this was a difficult task as the relationship was nonlinear. As such, the following concession was made for the Industry VMTI system when computing the PD: targets that were near enough to the targets in the frame were counted as hits. In quite a few cases, this admitted targets that were almost certainly false positives.

Although the implementation speed for the algorithm is recorded as 30 frames per second, or real time for this data, it should be noted that the input data is sampled and VMTI results only output approximately every 4 frames. The fourth column of Table 1 indicates the number of frames reported on per second. The other two systems report VMTI results on every frame.

The Industry VMTI has a collection of tunable parameters, but no effort was made to optimise them for this sequence. This was due to the problem with the frame synchronisation and the difficulty extracting meaningful information from the system, and then relating this information to parameter settings. However, we would expect the parameters to be fairly optimal as the data set used for this experiment was a part of a longer sequence supplied to the software company to develop the system. We should point out that the Industry VMTI system limits the speed of moving targets that can be detected via an upper and lower threshold. As a result, the system did not detect fast moving targets consistently in the sequence.

Another point of note is that the system would sometimes erroneously report multiple hits on the one target; these were manually corrected and did not affect the FAR or PD results. Finally, the video sequence we used did not contain passing clouds, which are apparent through much of the full sequence. The motion of passing clouds tends to increase the FAR for the Industry VMTI system dramatically. In contrast, the ARIA system has a significant amount of code devoted to the removal of such clutter so that it does not generate false alarms.

The ARIA System

The ARIA system [12] is designed to automatically flag moving targets to a waiting analyst for subsequent action such as zooming in, or other forms of investigation. The emphasis of the algorithm is to have a very low FAR while providing reliable cues for moving targets to the analyst. To this end, the ARIA system produces a visual display and text message in a log file as output of its VMTI process. An important feature of the system is that it is able to leverage the abilities of the analyst to detect targets from evidence gradually accrued in the display. Candidate targets are displayed in the output images in red and over time this tends to generate trails in the imagery that correspond to moving targets. It is very often clear from this display that the scene contains moving targets, readily distinguished from noise, well before an actual moving target is flagged (which the system does in yellow). Importantly then, in terms of a deployed system, this provides a much higher degree of usefulness than its lower PD would otherwise indicate. Moreover, once a target is detected, the system tends to stay locked on to it. The parameters of ARIA were tuned

using 8 minutes of video, and prior testing showed that ARIA performed just as well on 50 minutes of video never seen before, as it did on the 8-minute test sequence. The operational implication of this behaviour is that once suitable parameter values are chosen, they remain suitable for as long as the sensor footprint on the ground stays roughly the same.

As pointed out above, the ARIA system also handles obscuration of the scene by passing clouds, which broadens its usability considerably. However, this appears to have contributed to its long implementation time, which at present renders the system impractical in real situations. It is hoped that the port of the system to ADSS in the C language, plus incorporating any optimisations possible, will improve the implementation time considerably.

In terms of the FAR and PD results, there are two other points that should be mentioned. The system only considers the greater central region of the image for VMTI results; the rest of the image is blanked out by a border region where there is insufficient overlap between comparative frames to carry out VMTI. For the particular sequence processed, this meant that moving targets at the side of the image were not detected and this is reflected in the PD result. Finally, the system tended at times to count multiple targets as one; the long trails that are generated in the imagery tend to merge targets travelling in convoy. In practice, this may well not impact on the effectiveness of the algorithm in the field, but it does contribute to a lower PD. The trails would sometimes split, resulting in multiple detections for the one target; as with Industry VMTI these were ignored and did not affect the FAR or PD results.

The ADSS VMTI System

The ADSS VMTI system uses feature-tracking code, followed by registration and background modelling to predict moving target candidates in the scene. A subsequent Probabilistic Multi Hypothesis Tracker (PMHT) was used to remove noise and generate the list of actual targets. As the PMHT component is a MATLAB algorithm and is not yet implemented in ADSS, it is not reported herein (although it will be the subject of a future report). As can be seen from the table of results, the results compare favourably with the other algorithms, in particular with a good PD and an acceptable processing time (recorded on a twin processor using a parallel ADSS processing pipeline). The result for the FAR was caused by a single false alarm detected in the sequence; as mentioned above the parameters in the system could be adjusted to reduce the FAR at the expense of a lower PD.

In this study, ADSS VMTI was run on a dual processor machine, each a 2.6GHz AMD64 Opteron, with 8GB RAM. The KLT feature tracking module to produce frame registration information was run on every fourth frame, and the background was modelled on every frame by interpolating the registration results. VMTI results were reported on every frame, as indicated by the fourth column of Table 1. The ADSS processing pipeline was set up with three parallel pipes, as this was found give the best timing results. A significant proportion of processing time would normally be spent decoding and handling the *mpeg* image format, but for the purposes of this study the file format was converted to the native ADSS format before processing. Based on these results then, we can claim a real time implementation of VMTI using hardware less than 5K AUD. It should be noted that the implementation time for the PMHT is not included in this timing result however, as it is currently implemented in MATLAB and is not part of the ADSS system. It is anticipated, however, that the implementation time for the PMHT in ADSS would not contribute significantly to the timing results produced, as the PMHT currently runs in real time

even in MATLAB on a 1.8GHz processor (and typically an algorithm implemented in C is one to two orders of magnitude faster than the same algorithm in MATLAB).

There is currently an inherent lag in the ADSS VMTI algorithm caused by the need to generate a background model before moving target candidates are produced. In the current example, the baseline to generate the background model was 150 frames, or 5 seconds. In real applications, this lag may be unacceptable. A shorter baseline can be used (and would require further experimentation). On the other hand, an implementation of the system using GPUs is currently underway. It is expected that the significant savings in implementation time will effectively allow an implementation without lag, apart from an initial boot-strapping phase.

Finally, in the case of a tracked target becoming occluded, the PMHT is designed to wait until the target reappears while there is sufficient confidence in the measurements being collected. In certain cases the PMHT is able to provide tracking information through occlusions while the target moves behind objects such as trees, thus producing some remarkable tracking results. In terms of comparisons to ground truth however, we find in such case that the system finds a result where the ground truth has not, and these show up as false positives. We therefore decided to suppress such results for the purposes of this study. The PMHT will also occasionally produce a track that jumps from one target to another if they pass very close together. As the ground truth information is simply a count of the number of targets in each frame, and individual tracks are not uniquely identified (a more difficult experiment to undertake), this did not affect the FAR and PD results recorded for the system.

4.3 Summary of Comparative Analysis

Three different VMTI systems have been compared on the basis of FAR and PD for a five-minute video sequence containing small moving targets. The Industry VMTI system was found to have the lowest performance in terms of these two measures, in particular the high FAR level would not be acceptable for a useful deployed system. Although the Industry VMTI system has the fastest implementation time, it processes and outputs results for only approximately one in four frames. A system such as the ADSS VMTI therefore compares very favourably because it is outputting VMTI results for all the available frames at a near real time frame rate. The caveat with the ADSS VMTI system however is the lag in the system caused by the need to generate a background model, which is a computationally demanding task that we are presently working to overcome. The two other systems discussed herein would appear to hold more promise for a real time VMTI system with a FAR low enough to be useful in real applications.

5 Conclusion

In this report, a review of VMTI in ADSS was presented. The VMTI subsystem has been devised for video from moving sensors, in particular airborne urban surveillance video. As illustrated in this report, the paradigm of the moving sensor poses some unique problems as compared to the stationary sensor, which we have largely solved by registering video frames over a short temporal window. Our solution draws on a number of algorithms from the computer vision community, and combines them in a novel system. In particular, we leverage existing algorithm development in shape from motion, *e.g.*, the KLT feature tracking and RANSAC algorithms, and combine it with

established work for the static camera scenario, *e.g.*, background modelling and frame differencing. The solution provides positional and size information for any moving targets in a given video sequence, on a frame by frame basis. Moreover, given suitable parallel non-specialised hardware, the system allows a near real time solution to VMTI in ADSS.

Three different VMTI systems have been compared on the basis of FAR and PD for a five-minute video sequence containing small moving targets. The Industry VMTI system was found to have the lowest performance in terms of these two measures, in particular the high FAR level would not be acceptable for a useful deployed system. Although the Industry VMTI system has the fastest implementation time, it processes and outputs results for only approximately one in four frames. A system such as the ADSS VMTI therefore compares very favourably because it is outputting VMTI results for all the available frames at a near real time frame rate. The caveat with the ADSS VMTI system however is the lag in the system caused by the need to generate a background model, which is a computationally demanding task that we are presently working to overcome.

Our future work on VMTI will focus on the porting of the ARIA system into ADSS, with a view to speeding up the implementation time and possibly exploiting any advantages the sysem offers. Based on the results of the comparative analysis, it is likely that the system will offer considerably advantages in reducing the FAR, in particular in sequences where there is moving cloud cover. Future work will also focus on the implementation and refinement of a number of tracking algorithms in ADSS, including PMHT and particle filters, with a view to optimising these algorithms for our VMTI system.

References

1. S. Ali and M. Shah. COCOA - tracking in aerial imagery. *Proc. Int. Conf. on Computer Vision*, Beijing, China, October 2005.

2. P. Arambel, M. Antone, M. Bosse, J. Silver, J. Krant and T. Strat. Performance assessment of a video-based air-to-ground multiple target tracker with dynamic sensor control. *Signal Processing, Sensor Fusion, and Target Recognition XIV, Proc. SPIE* vol. 5809, pp.123-134, SPIE Bellingham, WA, 2005.

3. P. Arambel, J. Silver, J. Krant, M. Antone and T. Strat. Multiple-yypothesis tracking of multiple ground targets from aerial video with dynamic sensor control. *Proc. SPIE Defence and Security Symposium, Signal Processing, Sensor Fusion, and Target Recognition XIII*. Orlando, FL, April 12 - 14, 2004, SPIE Bellingham, WA, 2004.

4. S. Birchfield. KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker `http://www.ces.clemson.edu/~stb/klt/`.

5. D. Booth, N. Redding, R. Jones, M. Smith, I. Lucas and K. Jones. Federated exploitation of all-source imagery. *Proc. Digital Image Computing: Techniques and Applications* Cairns, Australia, December 2005, pp.243-250.

6. D. M. Booth, R. Jones and N. J. Redding. Detection of moving objects from an airborne platform. *Proc. Image Vision Conference New Zealand* Dunedin, New Zealand, November 2005, pp.255-260.

7. F. Bremond and G. Medioni. Scenario recognition in airborne video imagery. *Proc. Computer Vision and Pattern Recognition 1998: Workshop on Visual Motion,* Santa Barbara, June, 1998.

8. U. Braga-Neto and J. Goutsias. Automatic target detection and tracking in forward looking infrared image sequences using morphological connected operators. *33rd Conf. of Information Sciences and Systems,* March 1999.

9. J. B. Burns. Detecting independently moving objects and their interactions in georeferenced airborne video. *Proc. IEEE Workshop on Detection and Recognition of Events in Video (EVENT '01),* 2001, pp.12-19.

10. T. S. Caetano. *Graphical Models and Point Set Matching*. PhD Thesis, Universidade Fereral do Rio Grande do Sul, July 2004.

11. F. Campbell-West and P. Miller. Evaluation of a Robust Least Squares Motion Detection Algorithm for Projective Sensor Motions Parallel to a Plane. *Proc. of the SPIE Opto-Ireland 2005 Image and Vision Conference*, Dublin, Ireland, 5823, pages 225-236, 2005.

12. R. S. Caprari. Video Moving Target Indication (VMTI) for Airborne Surveillance of the Land Environment. *Proc. of Land Warfare Conference 2004*, Melbourne, pages 357-364, September 2004.

13. H. Cheng and D. Butler. Segmentation of aerial surveillance video using a mixture of experts. *Proc. Digital Imaging Computing: Techniques and Applications (DICTA 2005),* Cairns, December, 2005, pp.454-461.

14. I. Cohen and I. Herlin. Detection and tracking of objects in airborne video imagery. *Proc. Computer Vision and Pattern Recognition Workshop on Interpretation of Visual Motion*, Santa Barbara, June, 1998, pp.741-746.

15. I. Cohen and G. Medioni. Detecting and tracking moving objects in video from an airborne observer. *DARPA Image Understanding Workshop,* IUW98, Monterey, November 1998.

16. I. Cohen and G. Medioni. Detecting and tracking objects in video surveillance. *Proc. IEEE Computer Vision and Pattern Recognition 99*, Fort Collins, June, 1999, pp.319-325.

17. D. Comaniciu, V. Ramesh and P. Meer. Real-time tracking of non-rigid objects using mean shift. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, 2000, pp.142-149.

18. J. Dale, D. Scott, D. Dwyer and J. Thornton. Target tracking, moving target detection, stabilisation and enhancement of airborne video. *Airborne Intelligence, Surveillance, Reconnaissance Systems and Applications II, Proc. SPIE,* vol.5787, SPIE Bellingham, WA, 2005, pp.154-165.

19. S. Davey. Video moving target indication using PMHT. *The 2006 IEE Seminar on Target Tracking: Algorithms and Applications,* 7-8 March 2006, Austin Court, Birmingham, UK

20. D. Davies, P. Palmer and Mirmehdi. Detection and tracking of very small low contrast objects. *Proc. 9th British Machine Vision Conf.*, Sept., 1998, Southampton, UK, pp.599-608.

21. N. Dodd. Multispectral texture synthesis using fractal concepts. *IEEE Trans. PAMI*, pages 703-707, September 1987.

22. X. Dong and A. Jinwen. Independent moving target detection for aerial video surveillance. *Int. Conf. on Space Information Technology, Proc. SPIE,* vol.5985, 2005.

23. M. S. Figueiredo and A. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol.24, No.3, March 2002, pp.381-396.

24. N. Friedman and S. Russell. Image segmentation in video sequences: a probabilistic approach. *Uncertainty in Artificial Intelligence,* 1997.

25. Q. Gao, Y. Zhang and A. Parslow. The Influence of Perceptual Grouping on Motion Detection. *Computer Vision and Image Understanding,* Vol. 100, pages 442-457, 2005.

26. W. Grimson and S. Stauffer. Adaptive Background Mixture Models for Real-time Tracking. *Proc. IEEE Conference on Computer Vision and Pattern Recognition,* 23-25th June, 1999, pp.22-29.

27. W. E. L. Grimson, C. Stauffer, R. Romano and L. Lee. Using adaptive tracking to classify and monitor activities in a site. *Computer Vision and Pattern recognition,* 23-25th June, Santa Barbara, 1998, pp.22-31.

28. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, second edition*. Cambridge University Press, March 2004.

29. F. Hsieh, C. Han, N. Wu, T. C. Chuang and K. Fan. A Novel Approach to the Detection of Small Objects with Low Contract. *Signal Processing,* Vol. 86, pages 71-83, 2006.

30. P. J. Huber. Robust Statistics Wiley Series in Probability and Mathematical Statisics, John Wiley and Sons, 1981.

31. M. Irani and P. Anandan. A Unified Approach to Moving Object Detection in 2D and 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6), June 1998.

32. M. Irani and P. Anandan. Video indexing based on mosaic representations. *Proc. IEEE* 86(5), 1998, pp.905-921.

33. M. Irani, B. Rousso and S. Peleg. Computing occluding and transparent motion. *Int. Journal of Computer Vision,* vol.12, issue 1, pp.5-16, Feb., 1994.

34. S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld. Detection and location of people in video images using adaptive fusion of colour and edge information. *Proc. Int. Conf. an Pattern Recognition,* Barcelona, Spain, 2000.

35. R. Jones, D. Booth, P. Perry, N. Redding. A review of registration capabilities in the analysts' detection support system. Technical Report DSTO–TR–1632, DSTO Technical Report, 2005.

36. R. Jones, B. Ristic, N. Redding and D. Booth. Moving target indication and tracking from moving platforms. *Proc. Digital Image Computing: Techniques and Applications* Cairns, December 2005, pp.359-366.

37. D. Koller, J. Weber, J. Malik. Robust multiple car tracking with occlusion reasoning. *Proc. of European Conf. on Computer Vision,* 1994, pp.186-196.

38. E. H. Land. Experiments in colour vision. *Scientific American,* 200, 5, May, 1959, pp.84-99.

39. A. Liu, A. Haizhou and X. Guangyou. Moving object detection and tracking based on background subtraction. *Proc. SPIE* vol. 4554, 2001.

40. D. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing,* vol.22(2), pp.143-155, 2004-07-06

41. D. Magee. Tracking multiple vehicles using foreground, background and motion models. *Proc. EECV Workshop on Statistical Methods in Video Processing,* pp7-12, June 2002.

42. J. Margarey and N. Kingsbury. Motion Estimation using a Complex-Valued Wavelet Transform. *IEEE Transactions on Signal Processing*, 46(4), April 1998, pp.1069-1084.

43. Y. Ohta. *Knowledge-based interpretation of outdoor natural colour scenes.* Research Notes in Artificial Intelligence 4, 1985, Pitman.

44. G. Privett, P. Harvey, D. Booth, P. Kent, N. Redding, D. Evans, and K. Jones. Software Tools for Assisting the Multi-source Imagery Analyst. *Proc SPIE Aerosense*, 46(4), April 2003.

45. G. Privett and P. Kent. Automated image registration with ARACHNID. *Proc. Defense and Security Symposium*, 28-1st April, 2005, Orlando, Florida.

46. N. J. Redding. Design of the Analysts' Detection Support System for Broad Area Aerial Surveillance. Technical Report DSTO–TR–0746, DSTO Technical Report, 1998.

47. N. J. Redding, D. M. Booth and R. Jones. Urban video surveillance from airborne and ground-based platforms. *Proc. IEE Symposium on Imaging for Crime Detection and Prevention (ICDP 2005),* pages 79-84, London, June 2005.

48. B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.

49. J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. *Proc. 6th European Conf. on Computer Vision,* 2000.

50. H. Sawhney, Y. Guo, J. Asmuth and K. Kumar. Independent motion detection in 3D scenes. *Proc. IEEE Int. Conf. on Computer Vision,* pp. 612-619, Sept. 1999.

51. M. Shah and R. Kumar. *Video Registration*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2003.

52. J. Shao, S. K. Zhou. Robust appearance-based tracking of moving object from moving platform. *Proc. 17th Int. Conf. on Pattern Recognition, 2004.*

53. H. Shekarforoush and R. Chellappa. A multi-fractal formalisation for stabilisation, object detection, and tracking in FLIR sequences. *Proc. Int. Conf. on Image Processing,* vol.3, 2000, pp.78-81.

54. J. Shi and C. Tomasi. Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition. (CVPR'94)*, pages 593–600, June 1994.

55. C. Stauffer and W. Grimson. Adaptive background mixture models for real time tracking. *Proc. IEEE Conf on Computer Vision and Pattern Recognition,* vol.2, 1999, pp.246-252.

56. A. Strehl and J. K. Aggarwal. Detecting moving objects in airborne forward looking infrared sequences. *Machine Vision Applications Journal,* vol. 11, 2000, pp.267-276.

57. J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles.* Addison-Wesley, Mass., 1974.

58. R. Whatmough. *Private Communication.* Intelligence, Surveillance & Reconnaissance Div., DSTO Edinburgh, P.O.Box 1500, Edinburgh, SA 5111, Australia.

59. L. Wixson, J. Eledath, M. Hansen, R. Mandelbaum and D. Mishra. Image alignment for precise camera fixation and aim. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* pp.594-600, 1998.

60. H. Yalcin, R. Collins, M. J. Black and M. Hebert. A flow-based approach to vehicle detection and background mosaicking airborne video. *Proc. Computer Vision and Pattern Recognition,* vol.2, 20-25th June, 2005, pp.1202.

61. Y. H. Yang and M. D. Levine. The background primal sketch: an approach for tracking moving objects. *Machine Vision and Applications,* vol.5, pp.17-34. 1992.

62. A. Yilmaz, K. Shafique, T. Olson, X. Li, N. Lobo and M. A. Shah. Target-tracking in FLIR imagery using mean-shift and global motion compensation. *Proc. IEEE Workshop: Computer Vision Beyond the Visible Spectrum,* Hawaii, 2001.

63. A. Yilmaz, X. Li and M. Shah. Contour based object tracking using level sets. *Proc. of 6th Asian Conference on Computer Vision (ACCV),* South Korea, 2004.

64. A. Yilmaz and M. Shah. Automatic feature detection and pose recovery for faces. *Proc. Asian Conf. on Computer Vision,* Australia, January, 2002, pp.284-289.

65. A. Yilmaz and M. Shah. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.26, no.11, Nov., 2004, pp.1531-1536.

66. S. Zhou, R. Chellappa and B. Moghaddam. Visual tracking and recognition using appearance-based modeling in particle filters. Proc. Int. Conf. on Multimedia and Expo., Baltimore, MD, July 2003.

DISTRIBUTION LIST

Video Moving Target Indication in the Analysts' Detection Support System

Ronald Jones, David M. Booth and Nicholas J. Redding

	Number of Copies
DEFENCE ORGANISATION	
Task Sponsor	
ASC(DIGO)	1 (printed)
S&T Program	
Chief Defence Scientist	1
Deputy Chief Defence Scientist Policy	1
AS Science Corporate Management	1
Director General Science Policy Development	1
Counsellor, Defence Science, London	Doc Data Sheet
Counsellor, Defence Science, Washington	Doc Data Sheet
Scientific Adviser to MRDC, Thailand	Doc Data Sheet
Scientific Adviser Joint	1
Navy Scientific Adviser	Doc Data Sheet and Dist List
Scientific Adviser, Army	Doc Data Sheet and Dist List
Air Force Scientific Adviser	Doc Data Sheet and Exec Summ
Scientific Adviser to the DMO	Doc Data Sheet and Dist List
Information Sciences Laboratory	
Deputy Chief Defence Scientist Aerospace	Doc Data Sheet and Exec Summ
Chief, Intelligence, Surveillance and Reconnaissance Division	1 (printed)
Research Leader, Imagery Systems	1 (printed)
Head, Image Analysis & Exploitation	4 (printed)
Guy Blucher	1 (printed)
Dr David Booth	1 (printed)
Dr Robert Caprari	1 (printed)
Dr Tristrom Cooke	1 (printed)
Dr Gary Ewing	1 (printed)
Matthew Fettke	1 (printed)
Merrilyn Fiebig	1 (printed)
David I. Kettler	1 (printed)

Rodney Smith	1 (printed)
Bob Whatmough	1 (printed)
DSTO Library and Archives	
Library, Edinburgh	1 and Doc Data Sheet
Defence Archives	1
Capability Development Group	
Director General Maritime Development	Doc Data Sheet
Director General Land Development	1
Director General Capability and Plans	Doc Data Sheet
Assistant Secretary Investment Analysis	Doc Data Sheet
Director Capability Plans and Programming	Doc Data Sheet
Director General Australian Defence Simulation Office	Doc Data Sheet
Chief Information Officer Group	
Director General Information Services	Doc Data Sheet
Strategy Group	
Navy	
Deputy Director (Operations) Maritime Operational Analysis Centre, Building 89/90, Garden Island, Sydney Deputy Director (Analysis) Maritime Operational Analysis Centre, Building 89/90, Garden Island, Sydney	Doc Data Sheet and Dist List
Army	
ABCA National Standardisation Officer, Land Warfare Development Sector, Puckapunyal	Doc Data Sheet (pdf format)
SO (Science), Deployable Joint Force Headquarters (DJFHQ)(L), Enoggera QLD	Doc Data Sheet
SO (Science), Land Headquarters (LHQ), Victoria Barracks, NSW	Doc Data Sheet and Exec Summ
Air Force	
SO (Science), Headquarters Air Combat Group, RAAF Base, Williamtown	Doc Data Sheet and Exec Summ
Joint Operations Command	
Director General Joint Operations	Doc Data Sheet
Chief of Staff Headquarters Joint Operation Command	Doc Data Sheet
Commandant, ADF Warfare Centre	Doc Data Sheet
Director General Strategic Logistics	Doc Data Sheet
COS Australian Defence College	Doc Data Sheet
Intelligence and Security Group	
Assistant Secretary, Concepts, Capabilities and Resources	1

DGSTA, DIO	1
Manager, Information Centre, DIO	1
Director Defence Intelligence System Staff	1
Director Advanced Capabilities, DIGO	1
Defence Materiel Organisation	
Deputy CEO, DMO	Doc Data Sheet
Head Aerospace Systems Division	Doc Data Sheet
Defence Libraries	
GWEO-DDP	Doc Data Sheet
UNIVERSITIES AND COLLEGES	
Australian Defence Force Academy Library	1
Head of Aerospace and Mechanical Engineering, ADFA	1
Hargrave Library, Monash University	Doc Data Sheet
OTHER ORGANISATIONS	
National Library of Australia	1
NASA (Canberra)	1
INTERNATIONAL DEFENCE INFORMATION CENTRES	
US - Defense Technical Information Center	1
UK - DSTL Knowledge Services	1
Canada - Defence Research Directorate R&D Knowledge and Information Management (DRDKIM)	1
NZ - Defence Information Centre	1
ABSTRACTING AND INFORMATION ORGANISATIONS	
Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts, US	1
Documents Librarian, The Center for Research Libraries, US	1
INFORMATION EXCHANGE AGREEMENT PARTNERS	
SPARES	
DSTO Edinburgh Library	5 (printed)

Total number of copies: printed 22, pdf 25

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA				1. CAVEAT/PRIVACY MARKING
2. TITLE Video Moving Target Indication in the Analysts' Detection Support System			3. SECURITY CLASSIFICATION Document (U) Title (U) Abstract (U)	
4. AUTHORS Ronald Jones, David M. Booth and Nicholas J. Redding			5. CORPORATE AUTHOR Defence Science and Technology Organisation PO Box 1500 Edinburgh, South Australia 5111, Australia	
6a. DSTO NUMBER DSTO–RR–0306	6b. AR NUMBER 013-600		6c. TYPE OF REPORT Research Report	7. DOCUMENT DATE April, 2006
8. FILE NUMBER 2004/1084978	9. TASK NUMBER INT04/028	10. SPONSOR ASC(DIGO)	11. No OF PAGES 39	12. No OF REFS 66
13. URL OF ELECTRONIC VERSION http://www.dsto.defence.gov.au/corporate/reports/DSTO–RR–0306.pdf			14. RELEASE AUTHORITY Chief, Intelligence, Surveillance and Reconnaissance Division	
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT *Approved For Public Release* OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111				
16. DELIBERATE ANNOUNCEMENT No Limitations				
17. CITATION IN OTHER DOCUMENTS No Limitations				
18. DSTO RESEARCH LIBRARY THESAURUS Target Detection, Video Processing, Image Analysis				

19. ABSTRACT

This report presents a review of a video moving target indication (VMTI) capability implemented in the Analysts' Detection Support System (ADSS). The VMTI subsystem has been devised for video from moving sensors, in particular, but not exclusively, airborne urban surveillance video. The paradigm of the moving sensor, which is a typical scenario in defence applications (*e.g.*, UAV surveillance video), poses some unique problems as compared to the stationary sensor. Our solution to these problems draws on a number of algorithms from the computer vision community, and combines them in a novel system. It will provide positional and size information for any moving targets in a given video sequence, on a frame by frame basis. Moreover, given suitable parallel non-specialised hardware, the system allows a near real time solution to VMTI in ADSS.